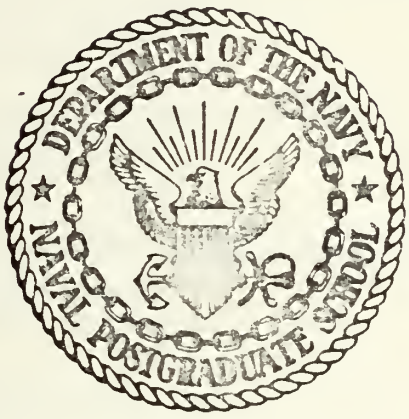
HARDWARE IMPLEMENTATION OF RECURSIVE FIXED-POINT DIGITAL FILTERS FOR MINIMUM QUANTIZATION NOISE

Carlos José de Almeida Rodrigues Rodolfo

NAVAL POSTGRADUATE SCHOOL
Monterey, California



THESIS

HARDWARE IMPLEMENTATION OF RECURSIVE FIXED-POINT DIGITAL FILTERS FOR MINIMUM QUANTIZATION NOISE

by

Carlos José de Almeida Rodrigues Rodolfo

September 1974

Thesis Advisor: S.R. Parker

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Hardware Implementation of Recursive Fixed-Point Digital Filters for Minimum Quantization Noise		5. TYPE OF REPORT & PERIOD COVERED Engineers Thesis; September 1974 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Carlos José de Almeida Rodrigues Rodolfo		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE September 1974 13. NUMBER OF PAGES 144
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Digital Filter
Arithmetic Quantization Errors
Hardware Implementation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Design and implementation of recursive digital filters with fixed point arithmetic using special hardware are considered in detail and applied to a mechanization of a second order filter structure with variable coefficients.

Two new methods of performing quantization after arithmetic operations within a digital filter are presented: quantization after addition and quantization before multiplication. Both

DD FORM 1473 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

(Page 1)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(20. ABSTRACT Continued)

methods are shown applicable to hardware implementation of digital filters and offer advantages over the usual quantization after multiplication. Error bounds are derived for these two quantization schemes and compared with the results previously obtained by other authors. It is concluded that the quantization before multiplication is the most suitable for hardware filter implementation. A design modification of the presently available hardware chips in order to permit round-off or truncation before multiplication is presented.

Hardware Implementation of Recursive Fixed-Point Digital Filters for Minimum Quantization Noise

by

Carlos José de Almeida Rodrigues Rodolfo
Lieutenant, Portuguese Navy
B.S.E.E., Naval Postgraduate School, June 1973
M.S.E.E., Naval Postgraduate School, December 1973

Submitted in partial fulfillment of the
requirements for the degree of

ELECTRICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
September 1974

ABSTRACT

Design and implementation of recursive digital filters with fixed point arithmetic using special hardware are considered in detail and applied to a mechanization of a second order filter structure with variable coefficients.

Two new methods of performing quantization after arithmetic operations within a digital filter are presented: quantization after addition and quantization before multiplication. Both methods are shown applicable to hardware implementation of digital filters and offer advantages over the usual quantization after multiplication. Error bounds are derived for these two quantization schemes and compared with the results previously obtained by other authors. It is concluded that the quantization before multiplication is the most suitable for hardware filter implementation. A design modification of the presently available hardware chips in order to permit round-off or truncation before multiplication is presented.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to express my appreciation to the Portuguese Navy for the opportunity to pursue this Electrical Engineer's degree program. I would also like to thank Dr. Sydney R. Parker for his guidance and motivation in this field and my wife, Ana Maria, for her constant help, encouragement and perserverence through all my scholarship period and in typing the smooth draft of this thesis.

Agradeço à Marinha de Guerra Portuguesa a oportunidade que me foi concedida permitindo-me terminar o curso de Engenheiro Electrónico. Agradeço também ao Professor Doctor Sydney R. Parker pela sua orientação e pelo interesse que criou em mim nesta area de estudo, e à minha mulher, Ana Maria, não só pela sua constante ajuda, encorajamento e perserverança durante todo o período escolar, mas também pela sua valiosa colaboração em dactilografar esta tese.

I. INTRODUCTION

A. IMPORTANCE AND APPLICATIONS OF DIGITAL FILTERS

A digital filter (D.F.) is defined [29] as a computational process or algorithm by which an input digital (discrete time and amplitude) signal or sequence of numbers is transformed into an output digital signal.

A digital filter can be compared to an analog filter as illustrated in Figure 1-1. A signal source x(t) is fed into the two processors. If the output y*(t) looks like the output $y_1^*(t)$ for all x(t), the upper and lower signal channels must be equivalent and then the digital processor is an equivalent of the analog filter, but operating on a digital signal, x*(t), from the analog to digital converter (ADC). Therefore the digital processor can be called a digital filter.

A digital filter can be implemented as a subroutine in a general purpose computer or as hardware in the form of a special purpose digital processor. In the hardware form, a D.F. is a collection of storage elements, adders and multipliers connected together in a prescribed way (filter structure), much as the continuous filter is an ordered connection of resistors, capacitors, inductors and active gain elements.

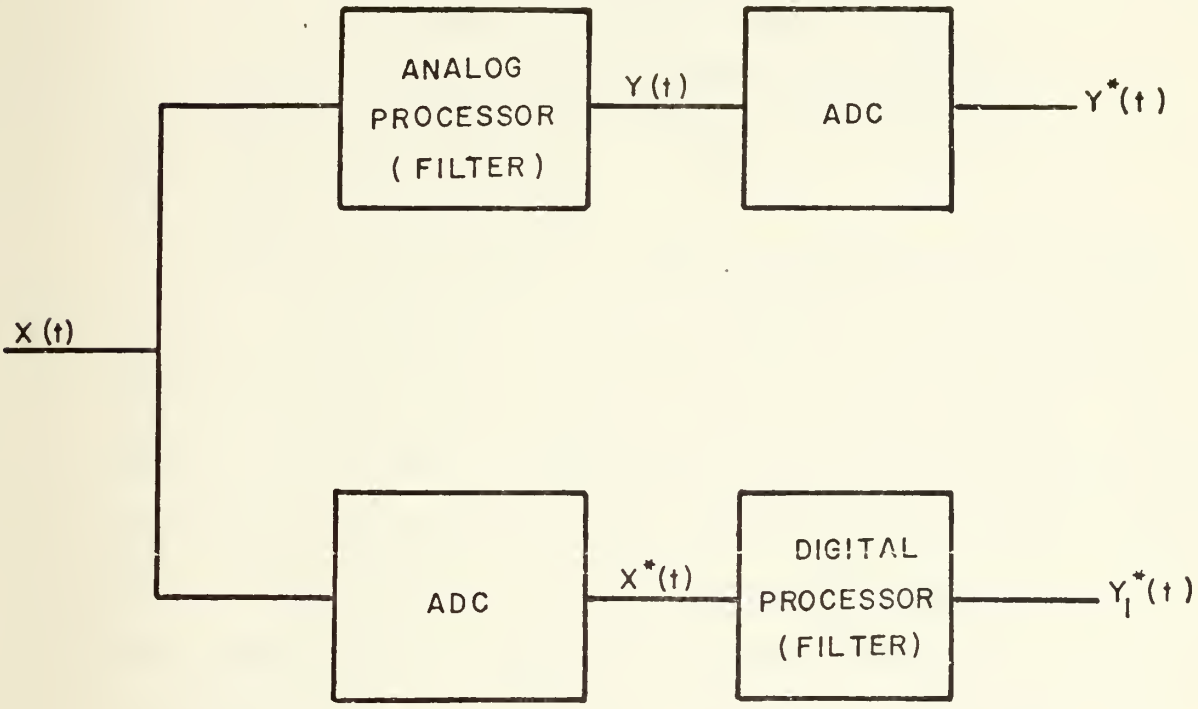


FIGURE 1-1 ANALOG AND DIGITAL FILTER COMPARISON

The advantages of digital filters over their analog counterparts are numerous [31]. Some of the advantages are:

a) arbitrarily high precision in the computational process,
b) no parameter or component value drifting,
c) flexibility in the processing procedure, which allows the construction of adaptive filters,
d) no necessity for impedance matching,
e) possibility to use time-sharing techniques,
f) easy realization of complex circuits,
g) high reliability,
h) small circuit size,
i) decreasing costs for mass-produced basic building blocks.

The following are typical examples of the superiority of digital filters over similar analog filter types: (1) Linear phase filters can be implemented by digital filters having extremely fast roll-off with either narrow or wide passbands or stopbands, and do not introduce nonlinear phase shift in the passband. (2) Comb filters are particularly useful for isolating repetitive signals of a known frequency. For example, in sonar systems, signals must be isolated from noise or other unwanted signals. (3) The extremely critical tolerances on crossover amplitude and phase characteristics of filters operating on adjacent passbands can be mechanized within any specified accuracy without drift or component aging effects. These accuracy and drift problems are encountered in spectrum analyzers and synthesizers having applications in radar, sonar, communications, and channel selectors. (4) Speech analysis and synthesis sometimes

requires a nonlinear phase response because both the magnitude and phase characteristics must be detected. In addition, the need to vary the filter characteristics is a necessity and may be varied or programmed easily with digital filters. (5) Two-dimensional filtering is widely used in the areas of image and geological data processing.

B. PREVIEW OF RESULTS

Digital filter implementation has been confined primarily to computer programs for simulation or for processing relatively small amounts of data, usually not in real time. However, the rapid development of integrated-circuit technology and specially large-scale-integration (LSI) is creating increasing interest in the hardware digital filter implementation. Mechanization hardware is discussed in Chapter II and its utilization in a digital filter design in Chapter III.

The design of a D.F. can utilize methods which are similar to those used for analog filters. Pole-zero analysis is essentially the same in the Z-domain used for discrete systems as it is in the Laplace transform domain used for continuous systems. Appendix A presents the Z-transform and the mapping of the s-plane into the z-plane, and discusses the significance of the pole positions. The transfer function decomposition methods of continuous systems are also easily applied to the Z-domain filter function and

result in the same filter forms, as shown in the discrete transfer function realization methods presented in Appendix B and in the functional transforms discussion in Appendix C. An example of a D.F. design using a Z-transform technique and its hardware implementation are illustrated at the end of Chapter III. A complex application of the North American Rockwell building chips in the hardware design of a second order section using a $SM_{11}{}^{T}$ structure and permitting variable coefficients and word lengths is presented in detail in Chapter IV.

Errors due to finite precision in the representation of numbers in a D.F. always occur. The quantization noise problem is particularly serious in recursive D.F. wherein the algorithm uses the results of previous calculations to generate present signal quantities. The fact that quantization errors are fed back can cause limit cycle oscillation. In Chapter V two new quantization methods are presented: quantization after addition (QAA) and quantization before multiplication (QBM). The former has been barely studied in the literature and the latter is not even mentioned. For the second order filter, using fixed point arithmetic, quantization bounds are derived for QAA and for QBM and compared with the results obtained by Yakowitz and S.R. Parker [20-32] for the case of quantization after multiplication (QAM). This study concludes that the bounds for QBM can be at most as large as the bounds for QAA and shows that the bounds for QBM are larger or equal to the bounds for QAA.

In Appendix D, using Lyaponov's direct method, a quantization bound for QAA in a two pole, no zero filter, is determined and compared with a value calculated in a previous work by Parker and Hess [1]. The result now obtained is half as large. Some other advantages of using QBM or QAA in hardware filter implementation are mentioned in the same chapter and a modification to the present hardware building chips is included in order to permit roundoff or truncation before multiplication in the implemented filter structure, otherwise restricted to truncation after multiplication.

II. DIGITAL CONSIDERATIONS

A. INTRODUCTION

A digital filter (D.F.) can be constructed from a small set of relatively simple digital circuits, primarily shift registers and adders, weel suited for large-scale integration (LSI) technology.

In this chapter the advantages of serial, two's complement binary arithmetic in the implementation of digital filters are discussed. The required shifting and arithmetic operations are described. Particularly, the serial/parallel multiplier and its circuits are studied in detail. The effect of sampling an analog signal is shown and a brief description of simple analog-to-digital and digital-to-analog converter circuits is also included.

B. TWO'S COMPLEMENT NOTATION

The 2's complement of a binary number is formed by simply subtracting each digit (bit) of the number from 1 and adding a one to the least significant bit (LSB). Two's complement coding of a digital number is used when both positive and negative numbers are to be represented. The two's complement of a number a, with N data bits, has the form

$$a_0\ a_1\ a_2\ a_3 \cdots a_N$$

where the bits a_i are either zero or one.

Since only fractional numbers will be used, the value of a has magnitude less than one, then

$$a = - a_0 + \sum_{i=1}^{N} a_i 2^{-i}$$

The bit a_0 is the sign bit and is commonly separated from the other bits by a decimal point, as represented in Figure 2-1, and the bit a_N is the least significant bit (LSB).

Positive numbers are coded in simple binary. Negative numbers are formed by taking the two's complement of the corresponding positive numbers.

1. Serial Processing

Serial processing of digital numbers is obtained by entering the digital number into sequential circuits one bit at a time with the least significant bit first. Parallel processing is accomplished if all bits are entered simultaneously. Gabel [30] has recently presented a parallel arithmetic structure for recursive digital filtering whose main advantage is a processing time independent of word length. Digital filters are generally serial machines since they present several advantages:

(i) They can be implemented using less and simpler hardware.

(ii) Carry-propagation delays found in parallel circuits are eliminated.

(iii) The delay operator z^{-1} of the digital filter is easily implemented with a single-input, single-output shift register.

(iv) Serial processing aids appreciably in the implementation of multiplexing schemes.

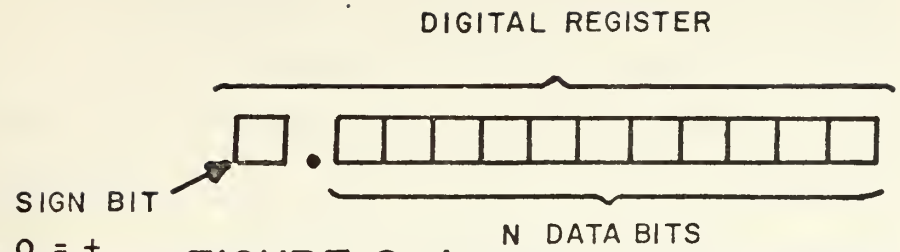


FIGURE 2-1 FORMATTING OF THE BINARY NUMBER

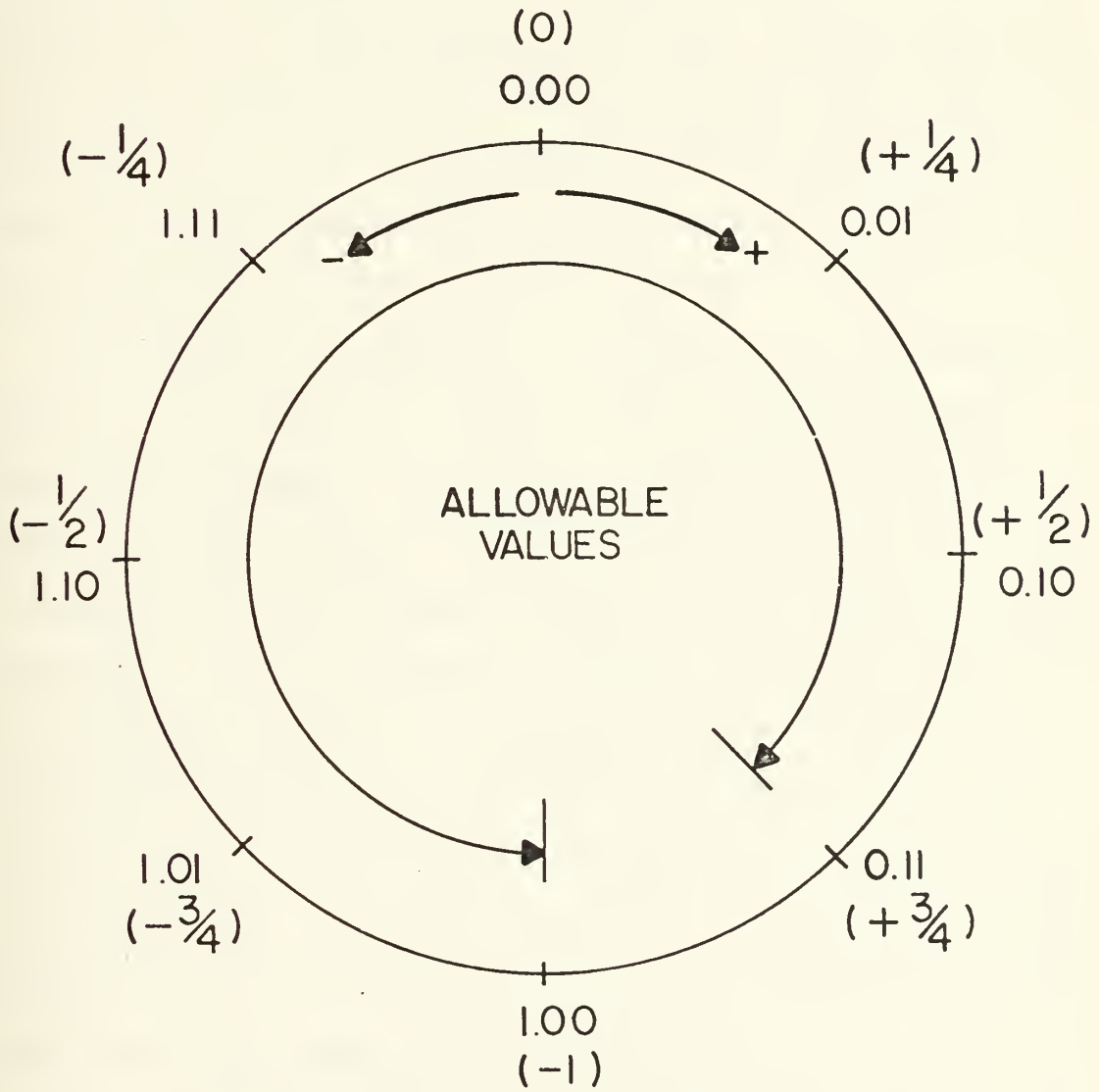


FIGURE 2-2 THE CYCLIC NATURE OF TWO'S COMPLEMENT ADDITION

2. Advantages of Two's Complement Notation

One advantage of two's complement is that formated data can be clocked into an arithmetic unit, with the least significant bit first, with no advance knowledge of the sign of the data [4]. Another advantage is associated with overflow in addition. Overflow in a digital filter occurs in the adder when the sum of the two numbers has a larger number of bits. Then the sum overflows into the sign bit. The output during overflow will be in error, but using two's complemented it can be recovered. If for instance, more than two numbers are being added, some of the partial sums will overflow, but the final sum may not.

The process of recovering an overflow is illustrated in Figure 2-2 in which the values of the two's complement number are arranged on a circle. Addition of positive numbers causes movement in the clockwise direction and that of negative numbers causes movement in the counter clockwise direction. Thus if positive overflow occurs the result will be a negative number and if negative overflow occurs the result will be positive. If +1/2 is added to +3/4, the result would be -3/4 due to overflow, but if a third number -1/2 were added, the result would be +3/4 which is correct. The same could be observed if one of the inputs has already overflowed from some previous operation.

The range over which the two's complement unit may be considered linear is from -1 to $(1 - 2^{-N})$ where 2^{-N} represents the least significant bit (LSB) and N the number of data bits in the number.

3. Number of Bits Required

The binary representation of a decimal number can have a very large length. Therefore, the number of bits necessary for representing a decimal number with a known accuracy has to be determined.

Let the decimal number

$$x = \sum_{j=1}^{D} b_j \, 10^{-j}$$

scaled such that $|x| < 1$, be known with an accuracy $(x-\Delta x) < x < (x+\Delta x)$ where

$$\Delta x = \frac{1}{2} 10^{-D}$$

and let the binary number (considering only the significant bits)

$$y = \sum_{i=1}^{B} a_i \, 2^{-i}$$

be the approximation of the decimal number, with an accuracy $\Delta y = \frac{1}{2} 2^{-M}$. Since the accuracy of the binary number has to be at least as great as the accuracy of the decimal number, it follows that

$$B \geq D \log_2 10 \simeq 3.32 \, D \qquad (2.1)$$

Therefore, the number of bits (sign bit excluded) necessary to represent in binary a decimal number (magnitude less than one) with an accuracy up to the D^{th} decimal place, is given by the first integer bigger than the product

$$3.32 \times 4 = 13.28 \ .$$

C. ARITHMETIC OPERATIONS

The only operations which have to be considered for a digital filter implementation are:

(i) Storage or shifting

(ii) Negation

(iii) Addition

(iv) Multiplication

1. Storage

Digital information is stored in a two state device called a flip-flop, which can remember, or store, a binary bit of information because of its bistable characteristic.

A shift register can be implemented using two such flip-flops placed in series and gated alternately as shown in Figure 2-3. Placing N shift registers cells in series the output is the input delayed by N clock periods.

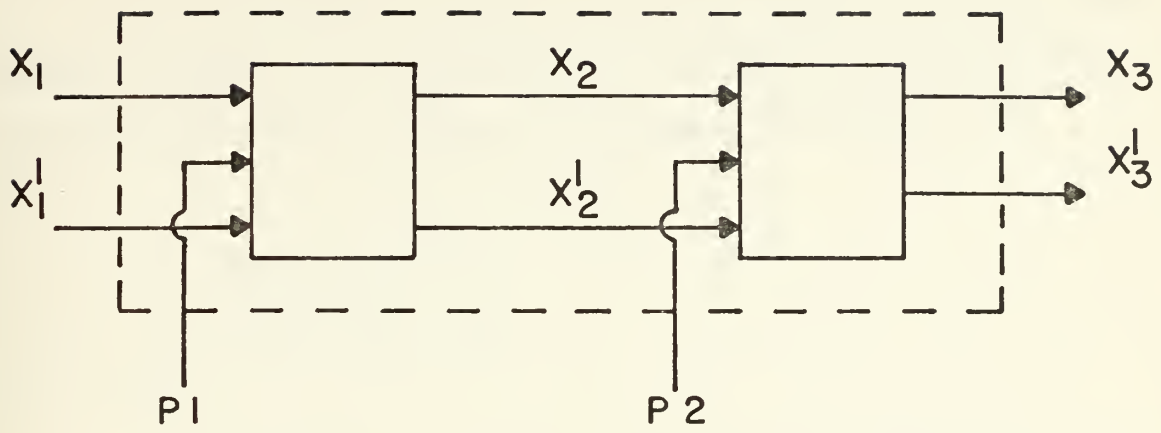


FIGURE 2-3 SHIFT REGISTER CELL

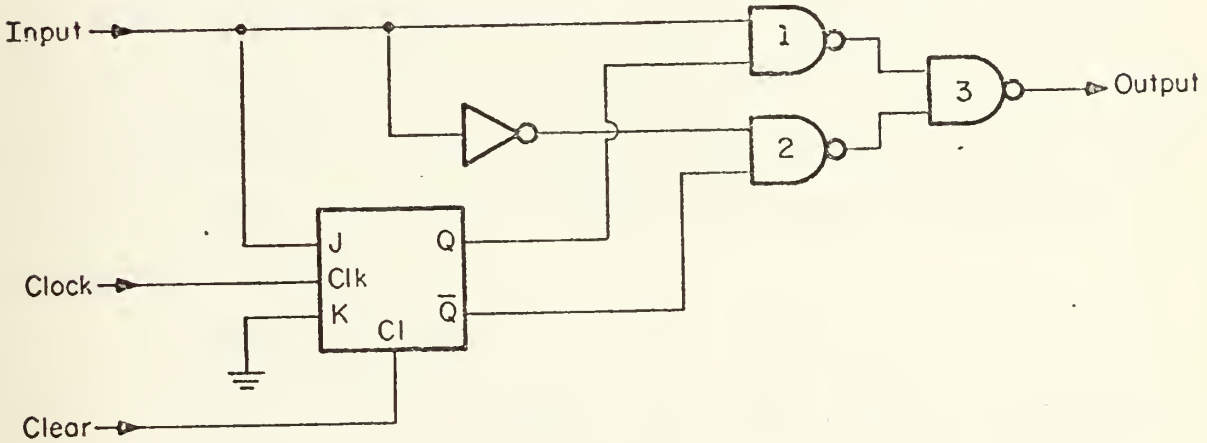


FIGURE 2-4 TWO'S COMPLEMENT INVERTER

2. Negation

A very useful method of inverting a two's complement number using serial arithmetic is to complement every bit which passes after, but not including, the first "1".

```
0. 1 0 1 0 1 0 0
Inverted
↓  ↓ ↓ ↓ ↓ ↓ ↓ ↓
1. 0 1 0 1 1 0 0
```

The sequential circuit presented in Figure 2-4 uses the method previously described for the implementation of a two's complement inverse. The input enters serially with the least significant bit (LSB) first with the Q output of the flip-flop initially cleared to zero. The bits pass unchanged through NAND gates 1 and 3. The first one will change the flip-flop state during the next clock pulse, thus all succeeding bits pass through the inverter and NAND gates 2 and 3. The clear pulse resets the flip-flop after the number has passed.

3. Serial Addition

Serial digital adders have three inputs (2 data and 1 carry) and two outputs (1 sum and 1 carry) as shown in Figure 2-5, and can be summarized by the truth Table II-1.

INPUTS			OUTPUTS	
A	B	C	1	2
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

TABLE II-1
TRUTH TABLE FOR SERIAL ADDER

From this truth table the following logic equations can be obtained

$$SUM = OUTPUT1 = AB'C' + A'BC' + A'B'C + ABC$$
$$= A(B'C' + BC) + A'(BC' + B'C)$$

$$CARRY = OUTPUT2 = A'BC + AB'C + ABC' + ABC$$
$$= BC + A(B'C + BC')$$

Figure 2-6 shows the logic implementation of the above equations.

In Figure 2-7(a) is shown a circuit used to implement two's complement addition involving one full adder and one flip-flop, which acts as the delay element. An inverter is used in the carry circuit of the standard full adder integrated circuit.

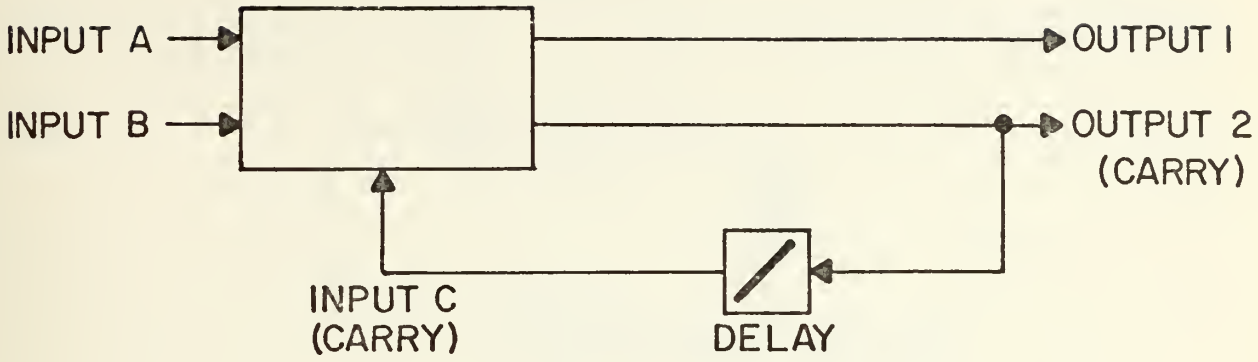


FIGURE 2-5 SERIAL ADDER

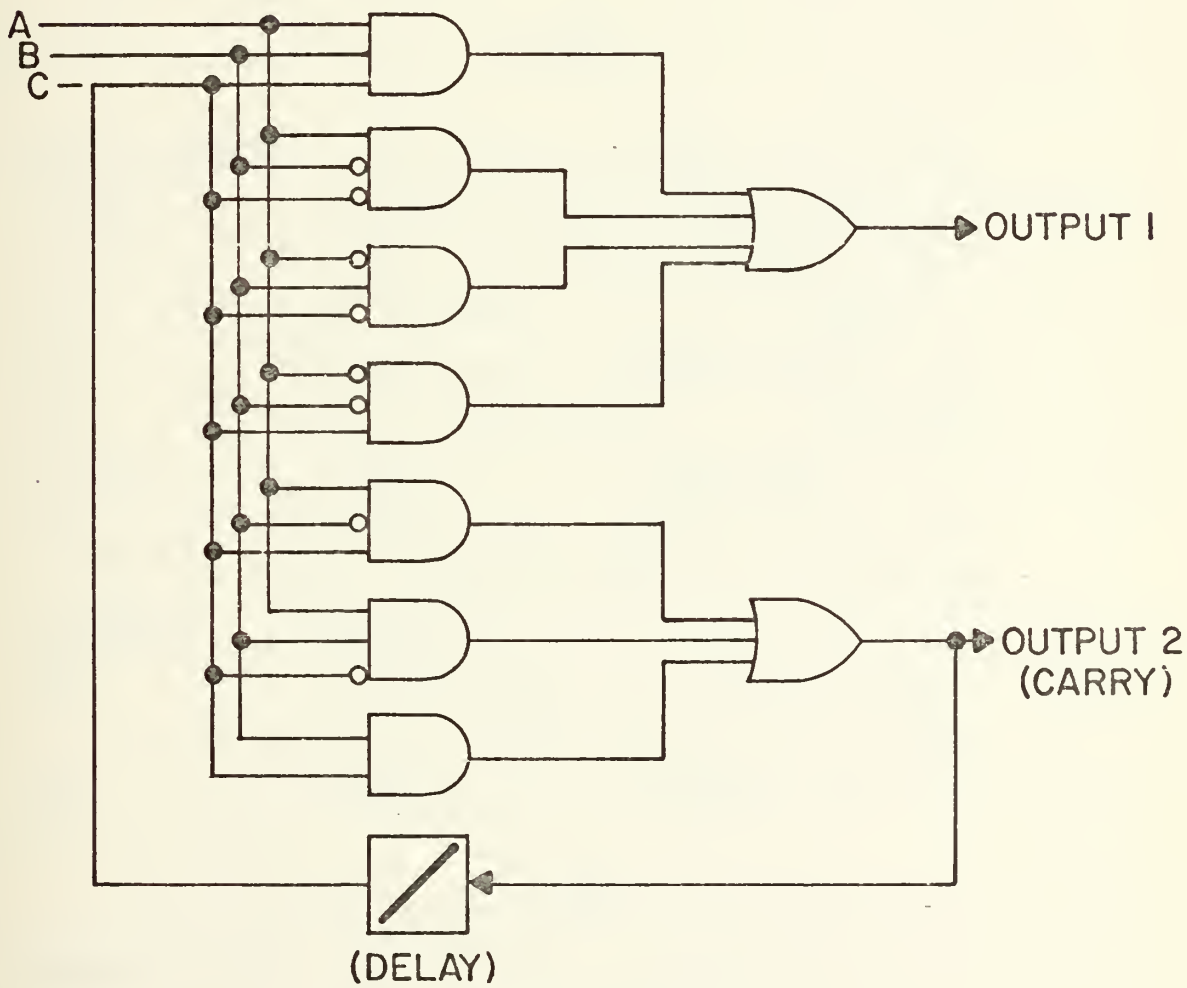
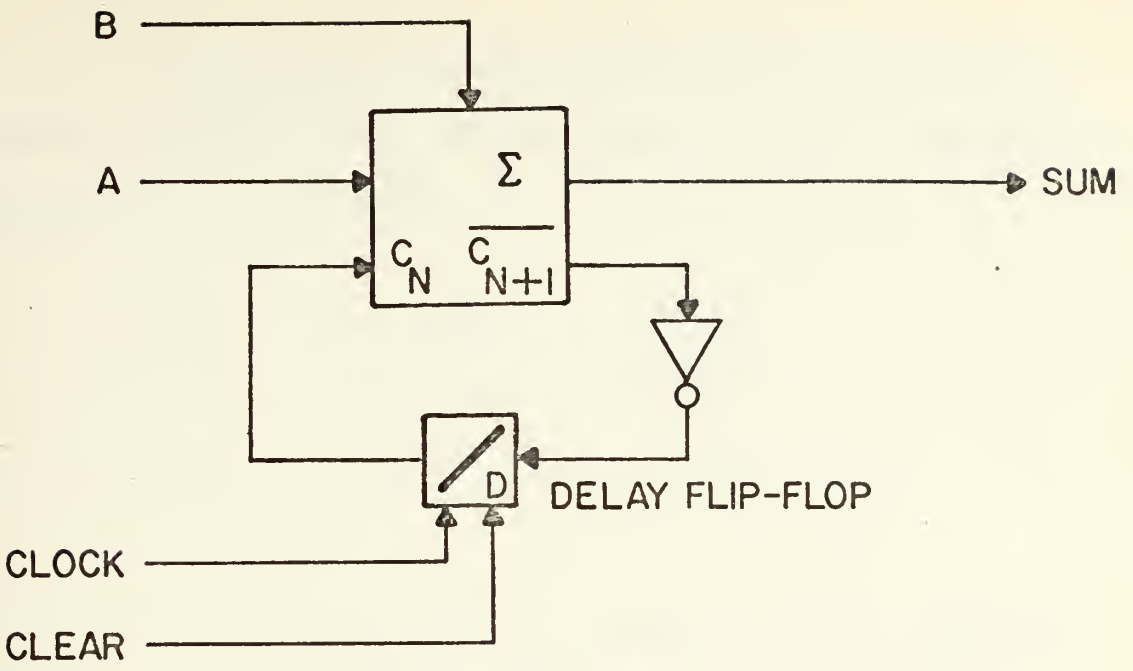
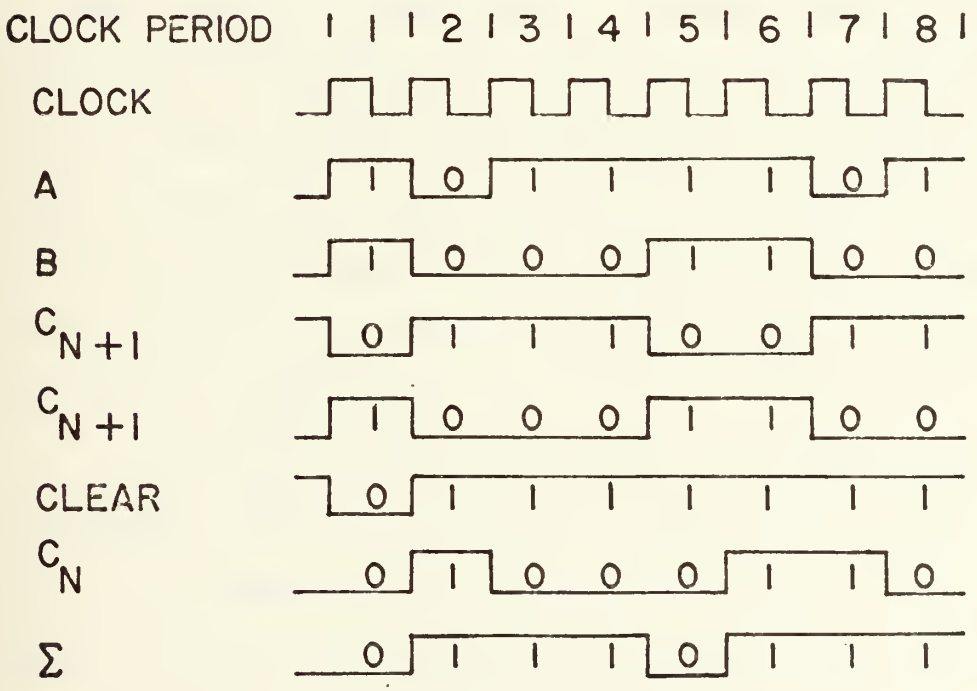


FIGURE 2-6 SERIAL ADDER LOGIC



(a)



(b)

FIGURE 2-7

(a) SERIAL ADDER

(b) TIMING DIAGRAM

To illustrate the operation of this circuit, an example of the addition of two numbers in two's complement notation will be performed.

A	1.0111101	(-67/128)
B	0.0110001	(+49/128)
A + B	1.1101110	(-9/64)

The corresponding timing diagram of this addition is shown in Figure 2-7(b). Assuming that the transfer information takes place when the clock changes from zero to one (positive going edge), it can be observed that during each clock period the full adder adds the bits A, B and C_n corresponding to that time and produces the sum Σ and the carry output C_{n+1}, this one will be delayed by one clock period so that it will appear at the input C_n during the next time period. A clear pulse will zero the carry during the first time period.

The time difference between the time the input bit enters and the time at which the output bit appears is called the "propagation delay" of the adder. The propagation delay to the sum output is usually larger than that of the carry output.

In order to avoid synchronization errors, flip-flops are generally necessary between adder stages to keep the data in synchronization.

4. Multiplication

Multiplication is the most complex and the most time consuming arithmetic operation required in digital filters. Normal binary multiplication is performed by successive additions and shifting, which process is controlled by the multiplier bits: if a 1, the multiplicand is added to the sum of partial product; if a 0, no addition is performed.

Since the filtering process must operate synchronously, the multiplication must be of fixed time duration. In addition to the speed considerations the amount and the complexity of the hardware required to perform multiplication is also important. Considering these factors, the serial/parallel multiplier (SPM), in which a serial data is multiplied by a parallel coefficent word, has been used almost exclusively.

The serial/parallel multiplier (SPM) accepts an M-bit serial multiplier and an N-bit paralled multiplicand input. Figure 2-8 shows a basic SPM, where a_1 represents the most significant bit (MSB) and a_n the least significant bit (LSB). The multiplier enters serially on the line "m" with the LSB appearing first. The number of adders in this SPM depends on the number of bits of the multiplicand. N-1 full adders are required for a N bit multiplicand. If a 1-bit appears on the multiplier serial input line, m, the stored multiplicand is gated to the adders through the AND-gates and the first partial product is generated. Each individual sum at each adder is then delayed 1-bit time and input to the next

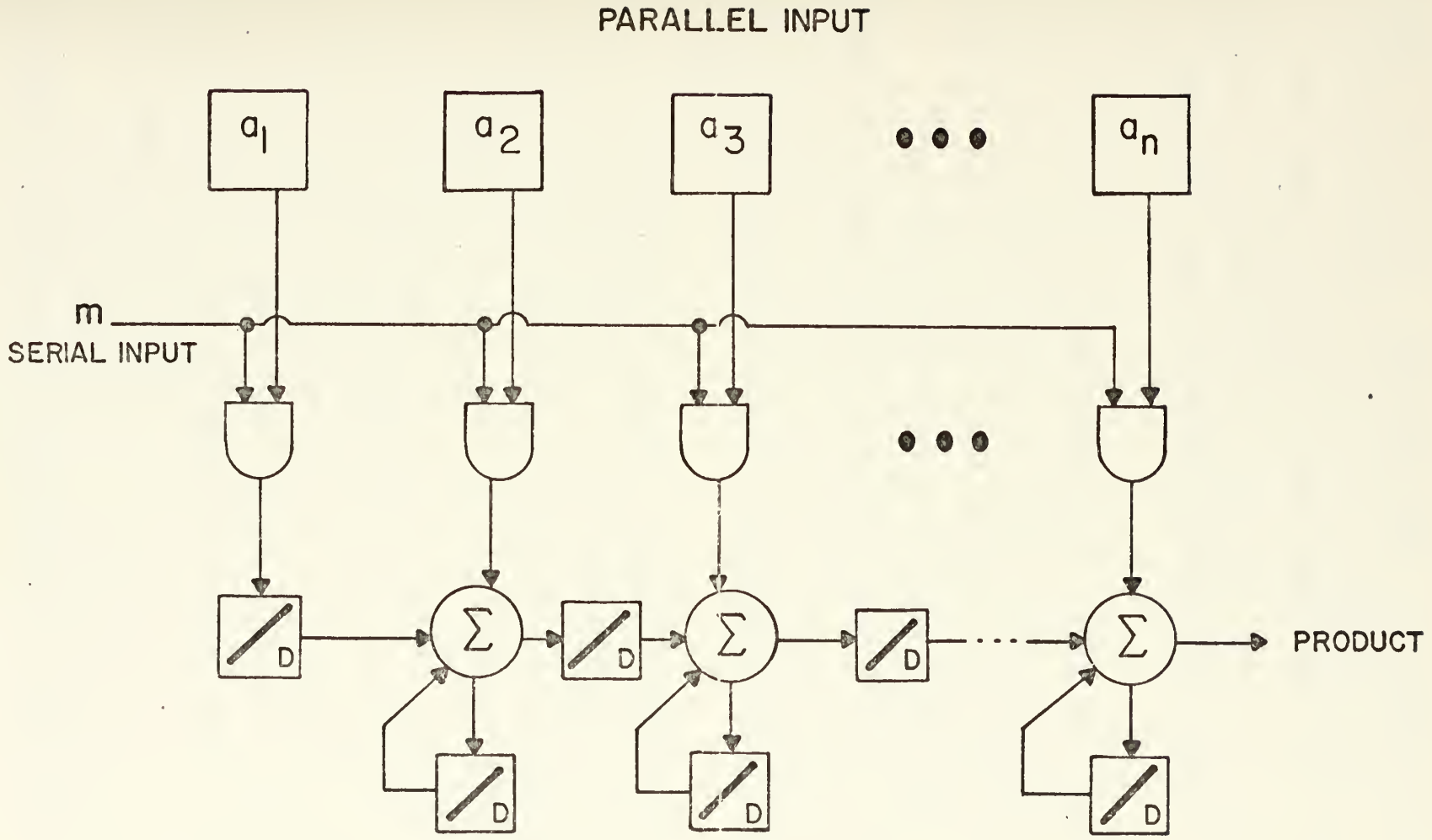


FIGURE 2-8 BASIC SERIAL / PARALLEL MULTIPLIER

adder. The carry from each adder is stored in the flip-flop which provides 1-bit delay so that the carry is fed back into the adders during the next clock time. If a "0" bit appears on the multiplier, causes all zeros to be sent to the adder and then the partial product will also be all zeros.

The LSB of the product will appear at the sum output of the last adder during the first clock period and the MSB will appear at the output during clock time N+M.

The modified version of the basic SPM shown in Figure 2-10 generally increases the versatility of the device, since it has the capability of multiplying either positive or negative numbers represented in two's complement coding.

The multiplication of a negative multiplicand with a positive multiplier is illustrated in Figure 2-9a. As before a "1" in the multiplier causes the multiplicand to be shifted to the left, but due to the negative multiplicand, the multiplicand sign-bit must be spread to perform the required correction. Thus the multiplier being "1", and the multiplicand negative (MSB is 1) 1's must be spread to the left of the MSB of the partial products. The multiplier being "0", the partial product will be all zeros, and 0's will spread to the left.

The multiplication of a positive multiplicand with a negative multiplier is illustrated in Figure 2-9b. In

```
          1.1 0 0 1 1      (-) MULTIPLICAND
          0.0 1 0 1 1
        ---------------
1.1 1 1 1 1 1 0 0 1 1
1.1 1 1 1 1 0 0 1 1 0
0.0 0 0 0 0 0 0 0 0 0
1.1 1 1 0 0 1 1 0 0 0
0.0 0 0 0 0 0 0 0 0 0
0.0 0 0 0 0 0 0 0 0 0
---------------------
1.1 1 0 1 1 1 0 0 0 1
```

(a) Two's complement multiplication of

$(+11x2^{-5})(-13x2^{-5}) = -143x2^{-10}$

```
          0.0 1 1 0 1
          1.1 0 1 0 1      (-) MULTIPLIER
        ---------------
0.0 0 0 0 0 0 1 1 0 1
0.0 0 0 0 0 0 0 0 0 0
0.0 0 0 0 1 1 0 1 0 0
0.0 0 0 0 0 0 0 0 0 0
0.0 0 1 1 0 1 0 0 0 0
1.1 0 0 1 1 0 0 0 0 0
---------------------
1.1 1 0 1 1 1 0 0 0 1
```

(b) Two's complement multiplication of

$(-11x2^{-5})(+13x2^{-5}) = -143x2^{-10}$

Figure 2-9

this case an ordinary multiplication will be performed except for the multiplier sign bit. The partial product of the multiplier sign bit has to be complemented, or since in this case the MSB of the multiplier is "1", the two's complement of the multiplicand is added instead to achieve the required correction.

In Figure 2-10 the network at the extreme left involving one AND-gate, one OR-gate and one type T flip-flop, acts as the sign spreader of the multiplicand as required. T_s is a single pulse, one clock period in length, which occurs at the time in which the sign bit of the multiplicand appears at the input. Therefore only the sign bit of the multiplicand is gated to the flip-flop. If the multiplicand is positive, the sign bit will be zero and this circuit will take no action. If the multiplicand is negative, the sign bit will be one and the T flip-flop, which was previously set to zero state by T_c, will change to one state and hold for the rest of the multiplication process. Therefore, the sign of the multiplicand will be spread. The time signal T_c is a single pulse occurring at the time the sign bit of the product appears at the output and its function is clear all flip-flop before the next multiplication.

T_0 is a single pulse occuring during the first time period of the multiplication process. The OR-gate in the carry circuit of the first adder and this time signal, T_0,

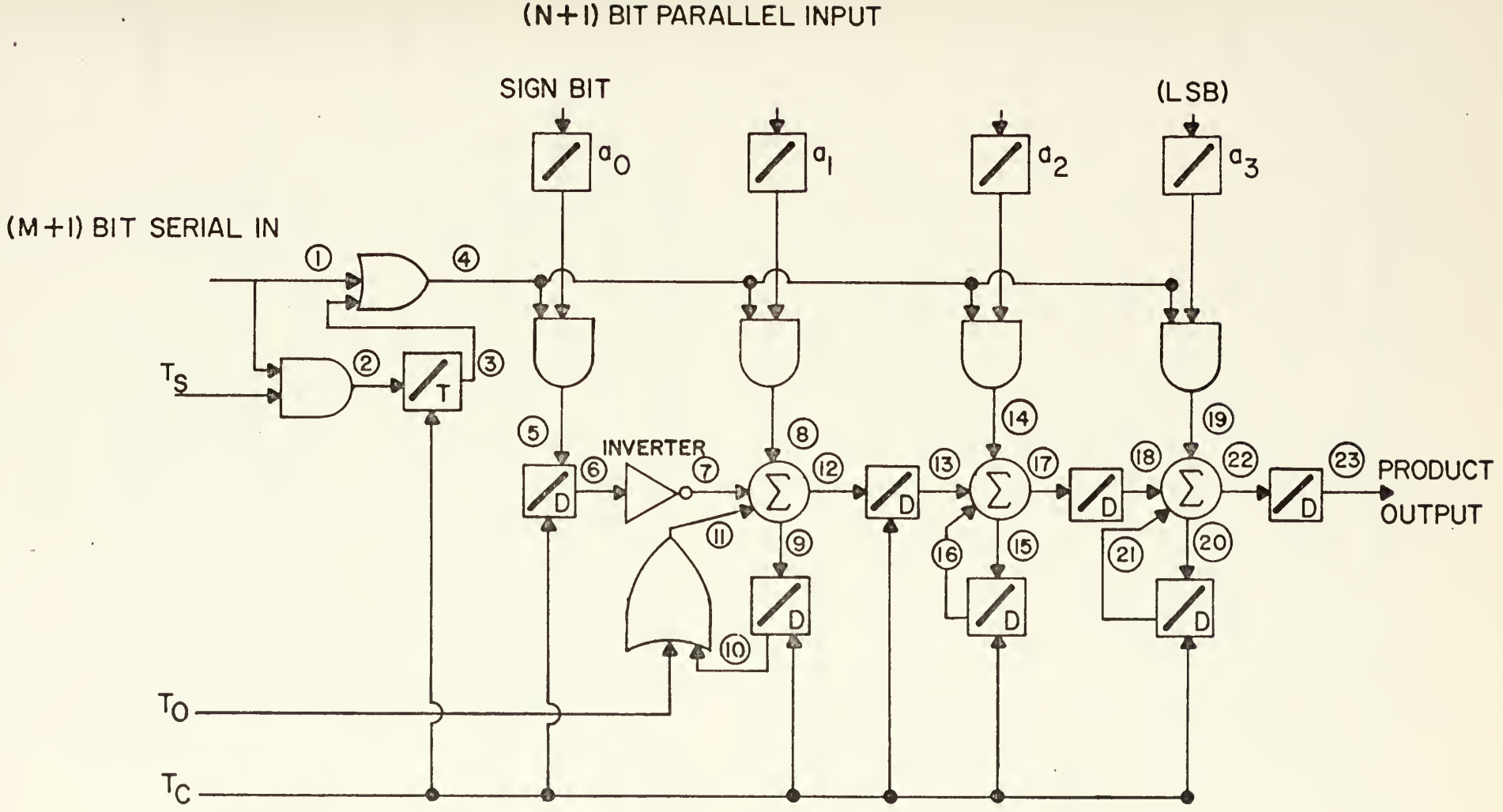


FIGURE 2-10 4-BIT SERIAL / PARALLEL MULTIPLIER

are used to subtract the multiplicand as required when the multiplier is negative. If the multiplier is positive, a_0 will be zero. Taking the 4-bit SPM of Figure 2-10, then point 5 will always be zero. The inversion after the delay will make point 7 one and its sum with 11 (which is one since T_0 at the input of the OR-gate is one at the first time period of the multiplication process), will generate a carry one at point 11. Therefore the output 12 of the first adder represents only the A input of the adder.

If the multiplier is negative, point 5 will depend on the existing multiplicand serial input bit during each time period. This circuit operates as two's complement subtracter for the multiplicand when the multiplier is negative.

The operations of the sign-spreader and the subtracter perform the corrective measure which enables the SPM to perform positive, negative and mixed multiplication.

An additional delay flip-flop included in the sum output of the last adder besides compensation for propagation delay, provides an extra delay required when two's complement multiplication is performed. When a N-bit number if multiplied by a M-bit number the resulting product has M+N+2 bits, but only M+N bits have magnitude information. The remaining 2 bits will indicate the sign of the product. The redundant sign bit can be eliminated by truncation.

In order to illustrate the operation of the SPM of Figure 2-10 the following example with a negative multiplicand and positive multiplier is used.

1.1 0 1 1 0	Multiplicand A = -5/16
0.1 1 0	Multiplier B = + 3/4
0 0 0 0 0 0 0 0 0 0	
1 1 1 1 0 1 1 0 0	
1 1 1 0 1 1 0 0 0	
0 0 0 0 0 0 0 0 0 0	
1.1 1 0 0 0 1 0 0	Product AB = -15/64

A timing chart for this multiplication is presented in Figure 2-11, which shows the states of each circuit point labeled in Figure 2-10 for each time period.

This multiplier can be expanded to accept any length serial multiplicand and parallel multiplier numbers [4], however the timing signals must be changed accordingly so that they occur in proper correspondnece with the serial input number and the product.

In a digital filter the multiplier numbers are the coefficients of the filter transfer function. If a fixed filter is used, the coefficient will remain unchanged and the multiplier bits can be hard wired. However if the coefficients are variables, external switches may be set to

CIRCUIT POINT	TIME PERIOD										
	1	2	3	4	5	6	7	8	9	10	
1	0	1	1	0	1	1	0	0	0	0	
2	0	0	0	0	0	1	0	0	0	0	
3	0	0	0	0	0	0	1	1	1	0	
4	0	1	1	0	1	1	1	1	1	0	
5	0	0	0	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	0	0	
7	1	1	1	1	1	1	1	1	1	1	
8	0	1	1	0	1	1	1	1	1	0	
9	1	1	1	1	1	1	1	1	1	1	
10	0	1	1	1	1	1	1	1	1	0	
11	1	1	1	1	1	1	1	1	1	0	
12	0	1	1	0	1	1	1	1	1	1	
13	0	0	1	1	0	1	1	1	1	0	
14	0	1	1	0	1	1	1	1	1	0	
15	0	0	1	1	1	1	1	1	1	0	
16	0	0	0	1	1	1	1	1	1	0	
17	0	1	0	0	0	1	1	1	1	0	
18	0	0	1	0	0	0	1	1	1	0	
19	0	0	0	0	0	0	0	0	0	0	
20	0	0	0	0	0	0	0	0	0	0	
21	0	0	0	0	0	0	0	0	0	0	
22	0	0	1	0	0	0	1	1	1	0	
23	0	0	0	1	0	0	0	1	1	1	PRODUCT
T_s	0	0	0	0	0	1	0	0	0	0	
T_o	1	0	0	0	0	0	0	0	0	0	
T_c	0	0	0	0	0	0	0	0	0	1	

Figure 2-11. Timing chart of a two's complement multiplication with multiplicand -5/16 and multiplier +3/4

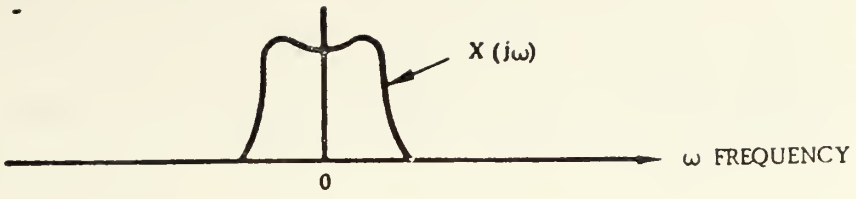
realize a particular filter - this is generally the case when laboratory units, or read-only-memory (ROM) are used - which is advantageous when the filter is to be multiplexed.

The advantages of using this two's complement serial/parallel multiplier for digital filter is now evident. There is only a N+1 bit delay (number of bits parallel input) and the multiplication process takes only M+N+2 time periods to be completed, but since the redundant sign bit can be truncated a word length of M+N+1 bits can be used. This type of multiplier using flip-flop between the full adders, eliminates greatly propagation delay problems.

D. SAMPLING

The sampling rate required for a sampler is determined by the analog input signal. If the input signal is periodic with period T, the minimum sampling rate which is called the "Nyquist rate" is 1/2T samples per second according to the sampling theorem.

Because of the effect of sampling, the original data spectrum is scaled and repeated across the entire spectrum. If the signal is sampled at a rate less than the Nyquist rate, or in other words, if the spectrum of the input signal is limited between $\pm\omega_s/2$, a distortion due to the overlaping side bands will occur, as observed in Figure 2-12b. This effect is called "folding" or "aliasing". Since the information lost by folding can not be recovered, care should be taken in the design of a digital filter. A practical limit



Narrowband Signal Before Sampling

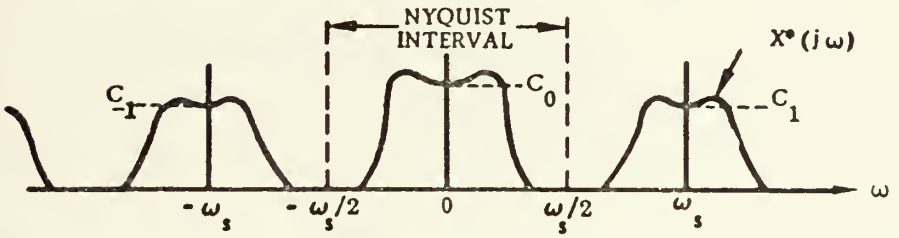
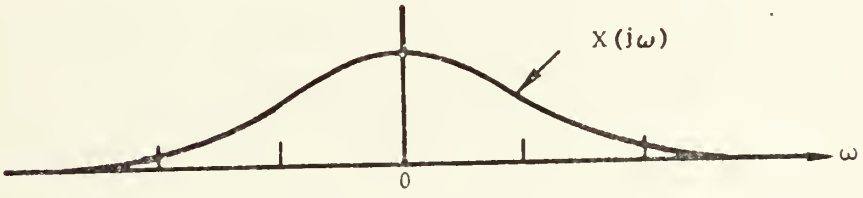


Figure 2-12a. Narrowband Signal After Sampling



Wideband Signal Before Sampling

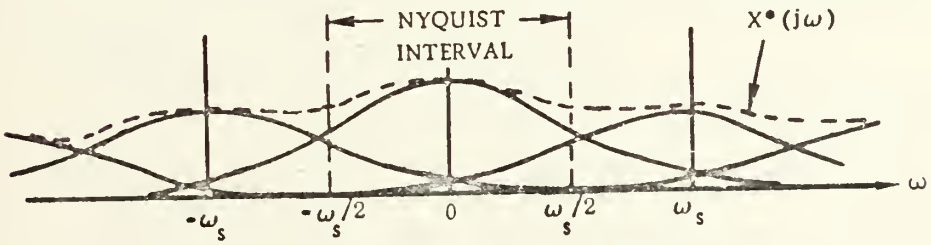


Figure 2-12b. Wideband Signal After Sampling

of $\pm\omega_s/5$ for the spectrum of the input signal has been found at the Naval Electronic Laboratory Center [13]. Therefore, digital filter applications are more suited for narrow band signals.

E. CONVERSION

1. Analog to Digital Conversion

The analog to digital converter (ADC) generates a digital number which is proportional to the amplitude of each pulse from the sampler by comparing the amplitude of input with some reference, which is generally generated by a digital to analog converter (DAC), as shown in Figure 2-13. The parallel inputs to the D/A come from an up/down counter which seeks a zero error at the comparator input. In order to hold the input constant during the conversion process it is necessary to precede the ADC by a sample/hold circuit, which holds the level sampled until the next sample is made. Since most ADC's have parallel outputs, as the one described, conversion must be made to a serial number, using a parallel-in serial-out shift register, before entering the digital filter.

2. Digital to Analog Conversion

The D/A conversion is generally a simpler process than the A/D conversion. The basic digital-to analog converter produces a certain output voltage for each different digital input. This is commonly done as shown in Figure 2-14, using a resistor network with one resistor connected to each bit of the input digital number. The resistor values are

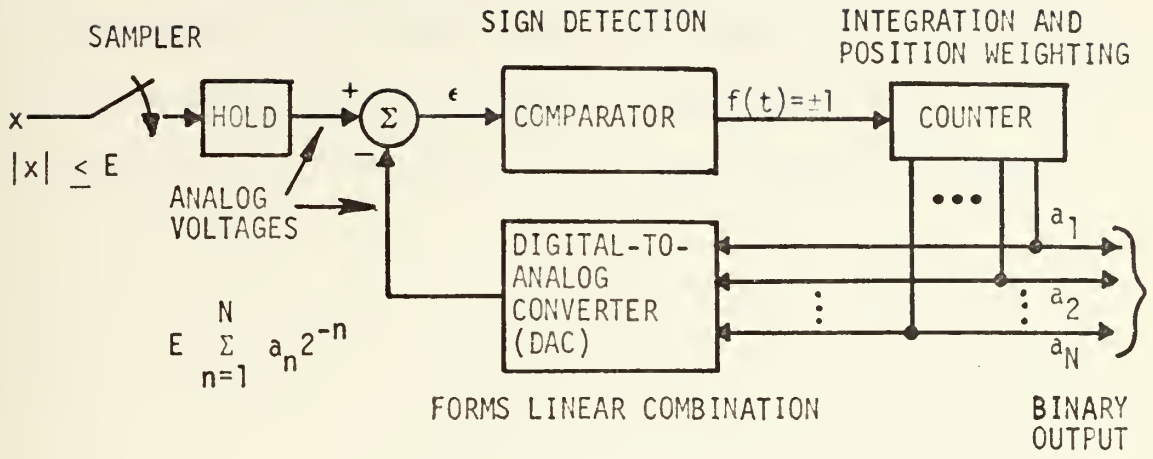
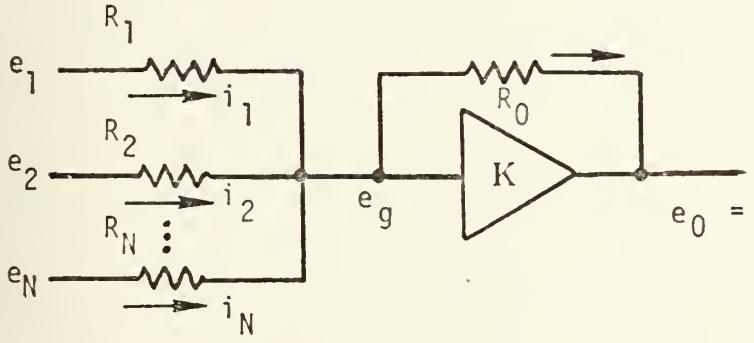


Figure 2-13. An Analog-To-Digital Converter

$$i = \frac{e_g - e_0}{R_0} = \sum_{k=1}^{N} i_k = \sum_{k=1}^{N} \frac{e_k - e_g}{R_k}$$



$$e_0 = Ke_g = \frac{-R_0 \sum_{k=1}^{N} e_k/R_k}{1 - \frac{1}{K} \sum_{k=0}^{N} R_0/R_k}$$

$$\simeq -R_0 \sum_{k=1}^{N} e_k/R_k$$

Figure 2-14. Current Summing with an Operational Amplifier to Obtain a Digital-to-Analog Conversion

weighted to be proportional to the value of each corresponding input bit. The resulting currents are then summed using an operational amplifier to produce a level which is proportional to the value of the input digital number.

III. DIGITAL IMPLEMENTATION. HARDWARE DESIGN CONSIDERATION

A. INTRODUCTION

The realization of a digital filter involves three main synthesis steps:

(i) Approximating the ideal filter transfer function by classical means and apply a convenient Z-transform technique [12]; an optimization algorithm to minimize, for example, a square error criterion in the frequency domain [26]; or any other direction design method to obtain a discrete filter which satisfies the given specifications.

(ii) Quantizing the multiplier coefficient of the filter in the appropriate cascade, parallel or hybrid form in such a way to minimize cost and complexity, while still satisfying the filter specifications.

(iii) Selecting a specific configuration for the digital filter, specifying the word length used and the arithmetic mode (only fixed point is being considered in this work), the quantization type (round off or truncation) and where in the circuit will be effective (generally after multiplication), so as to satisfy the specifications relating to quantization noise.

B. QUANTIZATION EFFECTS

When a D.F. is implemented with special purpose hardware (or on a computer) errors and constraints due to finite word length are unavoidable. This quantization effects must be

considered, both in deciding what word length (or register length) is needed for a given filter implementation and in choosing between several possible implementations of the same filter design, which will be affected differently by quantization.

There are four main errors due to quantization effects (i) Input quantization producing A/D conversion errors, (ii) Arithmetic quantization generating noise by the roundoff or truncation of quantities after arithmetic operations, (iii) Quantization of the filter coefficient producing a pole-zero displacement, and (iv) Constraints on signal levels imposed by the need of preventing overflow. The effects of these errors and constraints will vary depending upon the arithmetic used.

Weinstein and Oppenheim [22] have shown that floating point arithmetic is generally less noisy than fixed point arithmetic and it is known that floating point provides greater dynamic range. Fixed point mode is much easier to implement, and its error analysis is much less involved, therefore it is the one more often addressed in the literature. A discussion and bibliography of the literature concerning this error effects appears in [18-23-24]. The analysis of quantization noise due to roundoff after multiplication has been studied by stochastic [5-6] and deterministic methods [1-7-8-9], assuming uncorrelated noise sources. Under the general assumption of correlated noise sources a stochastic method has been studied by S.R. Parker, and P. Girard [25].

Mitra and Sherwood [21] have proposed a technique for estimation of pole zero displacement due to coefficient quantization in fixed point arithmetic. E. Avenhaus [27] has presented a method to find canonical structures which minimize the coefficient sensitivity due to rounding errors when small coefficient word length is used. Knowles and Olcayto [19] have indicated a method of analysis of the response of a D.F. affected by the coefficient accuracy using a "stray" transfer function in parallel with the corresponding ideal filter, but this method is not suitable for cascade realizations.

C. WORD LENGTH REQUIREMENTS

When a filter is constructed with digital hardware, the minimum word lengths needed for specified performance accuracy must be determined. This is one of the most important and difficult decisions in a digital design.

Figure 3-1 visualizes the relationship between the word lengths (number of bits in the number, sign bit excluded): in the input word (C), in the serial word being processed within the arithmetic unit (M) and in the multiplier coefficients (N). When the sign bit is included, these word lengths will be represented by C', M' and N', respectively.

1. Input Data Wordlength (C)

The input word length is the word length of the data out of the A/D converter. Therefore, it is related mainly to the input quantization error in the sampling A/D conversion

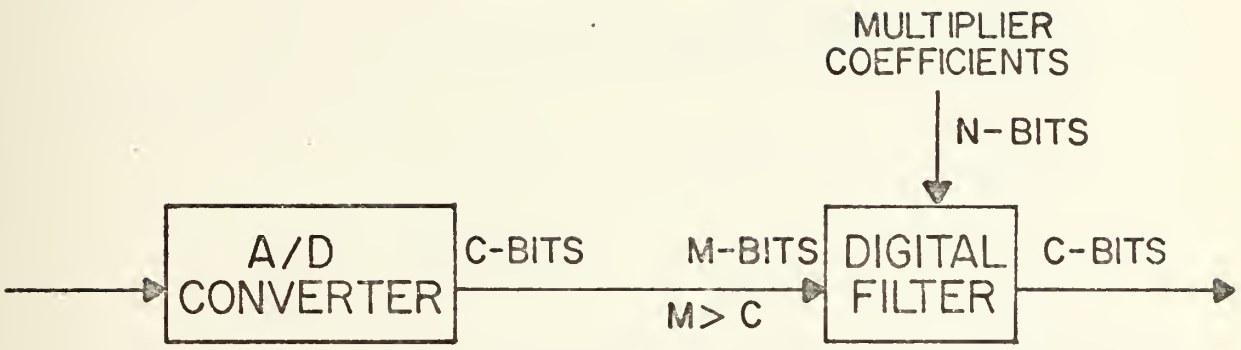


FIGURE 3-1 WORD LENGTHS IN A DIGITAL FILTER

process and determines the granularity or the number of levels of quantization required of the A/D converter.

The size of the quantization step used, h, depends principally on the dynamic range and on the granularity of the A/D converter. The dynamic range is the ratio between the largest signal or saturation level (x_{sat}) and the smallest signal detectable or threshold level (x_{th}).

Considering only the dynamic range dependence, the quantization step

$$h = x_{sat}/x_{th}$$

must be equal to the LSB with an accuracy of C significant bits, or

$$h = 2^{-C}$$

therefore

$$C = \log_2(x_{sat}/x_{th}) \tag{3.1}$$

Considering only the granularity of the A/D conversion, and assuming an additive white noise is introduced at the converter, resulting in a noise figure F, expressed in dB, the following equation can be obtained [3]:

$$C = \frac{F - 10\log_{10} \sigma_s^2}{20\log_{10} 2} \qquad (3.2)$$

where σ_s^2 represents the mean square level of the signal. As a design criterion, the signal may be assumed to have a Gaussian amplitude distribution with a standard deviation of 1/3, and then from equations (3.1) and (3.2) will result in

$$C' = C+1 = \max\{[1 + \log_2 \frac{x_{sat}}{x_{th}}],[\frac{F + 10\log_{10} 3}{20 \log_{10} 2}]\} \qquad (3.3)$$

2. Computational Data Word Length (M)

As mentioned previously, the arithmetic quantization noise is unavoidable and may be very significant in a D.F. and all the methods of analysis available presently are quite complex. Fettweis [17] has observed that round off (or truncation) noise depends only on the word length (M) at the input of the D.F., therefore M-C extra bits (all zeros initially) are appended to the A/D converter output.

The serial/parallel multiplier described later can handle any word length (M), however, if the coefficient word length (N) remains the same, the sampling rate and then the speed of the process will be reduced, as indicated by equation (3.8). Also the number of the shift registers used in the hardware filter implementation will increase as M increases as will be shown later.

3. Multiplier Word Length (N)

The multiplier coefficient length is associated with the accuracy with which the poles and zeros may be placed, or in other words, the tolerances of the filter design.

Multipliers with low sensitivity can be implemented with fewer bits, hence yielding a circuit with potentially lower cost and higher speed. Since first and second order sections are the building blocks being used, only the results of the coefficient accuracy applied to this case will be presented.

According to [3] a first order filter with a pole or zero $(s+\alpha)$ with a tolerance of $\pm\Delta\alpha$, requires a corresponding multiplier word length

$$N > \log_2 \left[2e^{-\alpha T} a^T \frac{\Delta\alpha}{\alpha}\right] \qquad (3.4)$$

and for a second order filter, with complex conjugate pair poles at $s = -\sigma_0 \pm j\omega_0$ with a characteristic equation is the z plane given by

$$1 + az^{-1} + bz^{-2} = (z - z_1)(z - z_2) = 0$$

where

$$a = -2r\cos\theta$$

$$b = r^2$$

$$z_{1,2} = r = e^{-\sigma_0 T}$$

$$\text{Arg } z_{1,2} = \theta = \omega_0 T$$

For the tolerances of $\pm\Delta\sigma$ and $\pm\Delta\omega$ the word length of the coefficient multipliers has to be:
for a:

$$N \geq -\log_2\left[4\sqrt{b}\,\frac{\Delta\omega}{\omega_0}\,\omega_0 T \sin(\omega_0 T)\right] \qquad (3.5)$$

for b:

$$N \geq -\log_2\left[4\,\sigma_0 T\, e^{-\sigma_0 T}\,\frac{\Delta\sigma}{\sigma}\right] \qquad (3.6)$$

As will be observed later the number of serial/parallel multipliers used will depend on this word length (N).

D. GAIN SCALING

Overflow occurs when a D.F. computes a number that is too large to be represented in the arithmetic used in the filter. If no compensation is made for the overflow, then large errors in the filter output will result.

Several techniques are used to compensate or to avoid overflow. One method is to detect overflow and then compensate for it immediately after it occurs. If a positive

overflow is detected, a large negative number is injected into the filter and if a negative overflow is detected, a large positive number is injected. The overflow will then be compensated due to the cyclic nature of 2's complement arithmetic, and no error will occur. Another method is saturation arithmetic where a sum that is too large to be represented is set equal to the largest representable number in the filter. The output will be in error, but it will avoid overflow oscillations.

The most common method of preventing overflow is the process of scaling. The simplest form of scaling is effectively to reduce the size of the input signal. However, if the analog input is reduced, the signal-to-noise ratio will usually be decreased. Therefore, it is usually more desirable to reduce the digital input signal with a scaler between the A/D converter and the filter input. This scaler can be a shift register which effectively divides by powers of two or a multiplier whose coefficient is less than one. This last approach will be the one used. In fact, all second order filter sections will be preceded by a scaling multiplier (K) that will be set just low enough to prevent overflow at any adder. Thereby, linearity is assured while maximizing the dynamic range of each section and consequently of the filter. This is achieved by seeking a value of K such that for all the possible digital filter inputs, $X(z)$, the output of each adder, $Y_1(z)$, will satisfy

$$\left|\frac{Y_i(z)}{X(z)}\right|_{z_{max}=\exp(j\omega T)} \leq 1 \qquad (3.7)$$

E. TIMING

Timing is another requirement in digital filter design, since sequential circuits are used. The "filter word" length (number of time periods required to process one input word before the next word may be entered) has to be determined. Mathematically the filter word length corresponds to the delay operator z^{-1} which appears in the desired D.F. transfer function. As will be shown in the examples presented later, the filter word length is a function of the multiplication time and it is generally given as (M' + N') bits, where M' and N' are respectively the number if bits used to represent the computational data word and the scaling coefficient in the multiplier (sign bit included). Then, the maximum word rate (sampling rate) at which the filter can operate is

$$f_W = \frac{f_B}{M' + N'} = \frac{f_B}{M + N + 2} \qquad (3.8)$$

where f_B is the bit rate, determined by the system clock rate, and (M' + N') is generally referred to as the word time.

F. HARDWARE DESIGN

The following discussion on hardware implementation will be restricted to MOS/LSI[1] technology. Two types of MOS/LSI chips developed by the North American Rockwell Microelectronics Company (NRMEC) will be presented and a design method of second order filter sections will be introduced. This method will be illustrated with a low pass digital filter example using a z-transform technique.

1. The Devices

The North American Rockwell Microelectronics Company (NRMEC) has developed two LSI processing devices to operate on two's complement formatted serial digital data and LSI compatible analog-to-digital and digital-to-analog converters. Table III-1 presents the characteristics of this MOS/LSI digital filter building block. Filters may be configured using this device over the frequency range of 0 to 20 KHz.

The serial/parallel multiplier (SPM) and the shift register adder (SRA) are the processing devices. This MOS/LSI device utilizes p-channel enhancement mode transistors. A four phase clock scheme is required to perform both the SPM and the SRA.

a. Serial/Parallel Multiplier (SPM)

One SPM chip forms the sign-corrected product of an input data word of any length and a scaling coefficient

---

[1]MOS technology refers to a device with three layers: metal-oxide-semiconductor. LSI means large-scale integration process.

Characteristics	SPM	SRA	A/D-D/A
Size (in mils)	142 x 136	180 x 216	180 x 180
Frequency (MHz)	1.5	1.0	1.0
Power Dissipation (in mw at 1 MHz)	35 max	200 max	75 max
Output Drive Capability	100 pf	50 pf	100 pf
Voltage (clock, input, supply	-30V max	-30V max	-30V max
Number of Devices(MOSFETS)	640	1250	1800
Mechanized terms	322	410	11 bit
Number of Pins (flat pack)	42	42	42

Table III-1. Characteristics of LSI digital filter devices from North American Rockwell Microelectronics Company

of length up to 8 bits plus sign. Longer coefficient multiplications can be performed by cascading SPM chips. The scaling coefficient (multiplicand) can be loaded in parallel or serial and transferred to parallel holding register. Generally in digital filters applications the scaling coefficient is input serially at SI1, least significant bit (LSB) first, by changing the TRS input from "0" to "1" one bit after inputting the sign bit, as observed in the timing diagram of Figure 3-3. The serial word (multiplier) is inputted LSB first into MI1 input and input TSS should be taken to a "1" for one bit at the same time as the sign bit appears on the MI1 input. The TMR signal being "1" clears the adders and sign bit circuitry and holds the output to "0". The LSB of the multiplier should be inputted 2 bits after this TMR signal.

From Figure 3-2 can be observed that the LSB of the product appears at the output (SO1 or SO2) one bit after the LSB of the multiplier input signal enters the MI1 input. For an N' bit coefficient multiplicand, the multiplication process will produce a delay of N' bits at the SO1 output. In Figure 3-3 a 9 bit delay between the sign bit of the multiplier input and the product output is observed for the 9 bit (8 + sign) scaling coefficient (multiplicand) used.

The multiplier performs proper sign connection only if the inputs (data and scaling coefficients) have

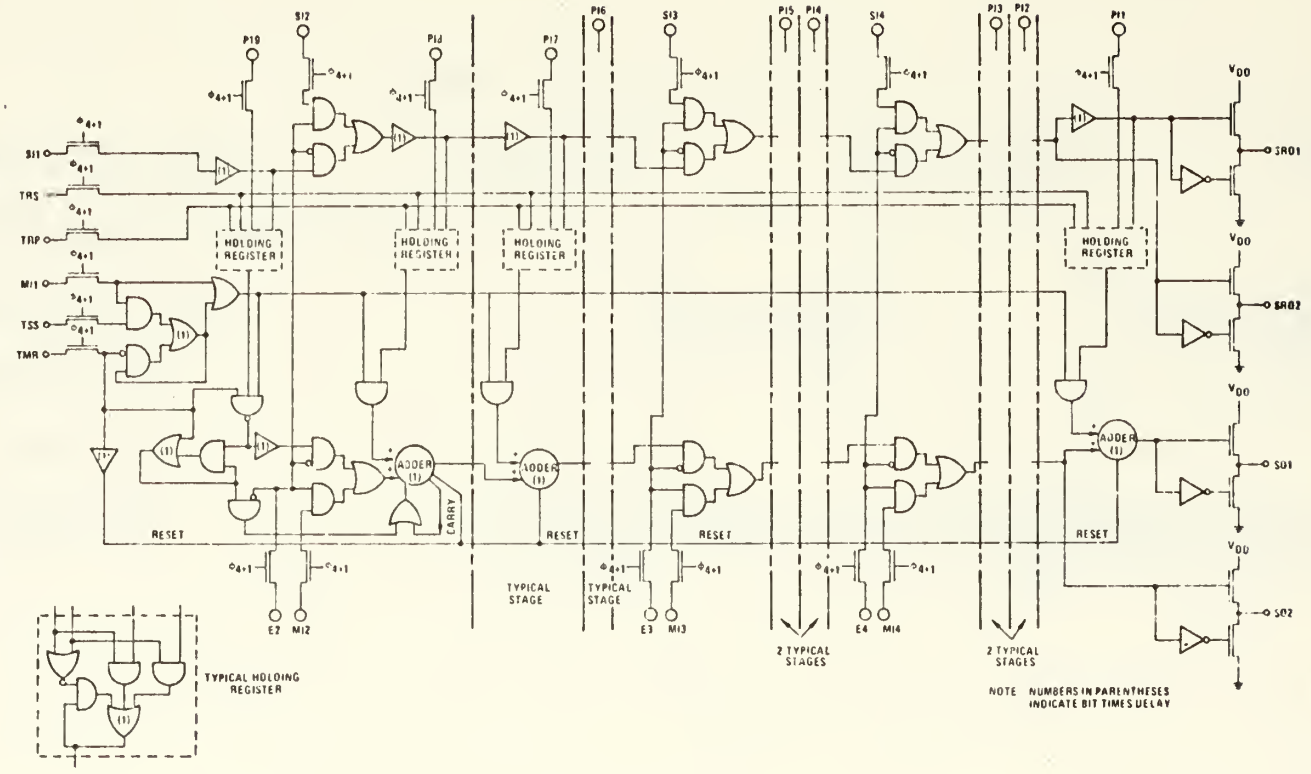


Figure 3-2. Block Diagram of 65001NA Serial/Parallel Multiplier

65001NA

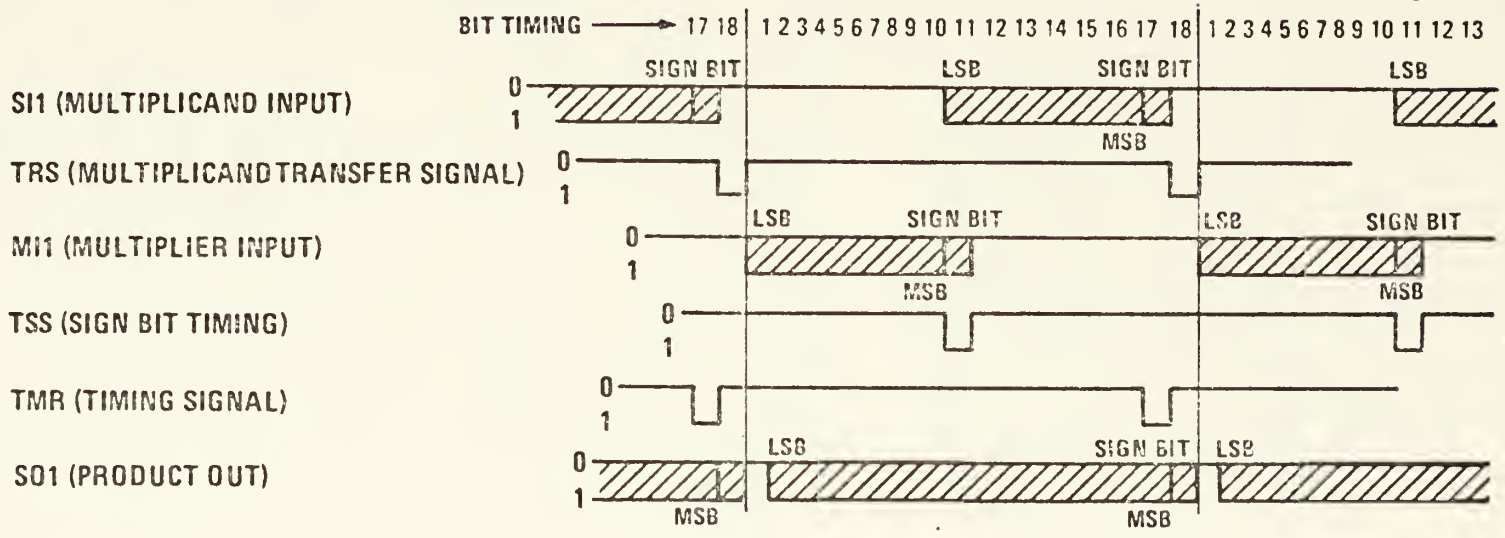


Figure 3-3. Timing Diagram for 8-Bit-Plus-Sign Multiplier and Multiplicand in Minimum-Time Cyclic Operation

magnitudes both greater than unity. This potential problem can generally be solved in a practical mechanization as will be shown.

b. Shift Register Adder (SRA)

As shown in the block logic diagram of Figure 3-4 and in the simplified functional diagram of Figure 3-5 a SRA consists of two identical 7 to 15 bit shift-and-hold registers, two 4-input adders and a timing and control circuitry.

Each adder exhibits a one-bit time delay. One of the adders is able to inhibit two inputs if the input CNI is made "1". Both adders are reset by a "1" on control inputs TR1 and TC21.

The register section is able of adjust in length to accommodate the length of the data word in the computational loop, by coding the inputs A, B and C. A shift register longer than 15 bits is obtained by cascading these register sections. Particular, a delay up to 30 bits can be obtained cascading the two sections of a single SRA chip.

The timing and control section provides the proper timing signals not only to the SRA but also to the multipliers that may be associated with that SRA. The timing signals T_1 and T_2 are the only required timing inputs.

2. Canonic Realization of Second Order Sections

Given a linear time invariant system it is shown in Appendix B that its transfer function can be expressed as a parallel, cascade or hybrid realization of first and second order transfer function sections.

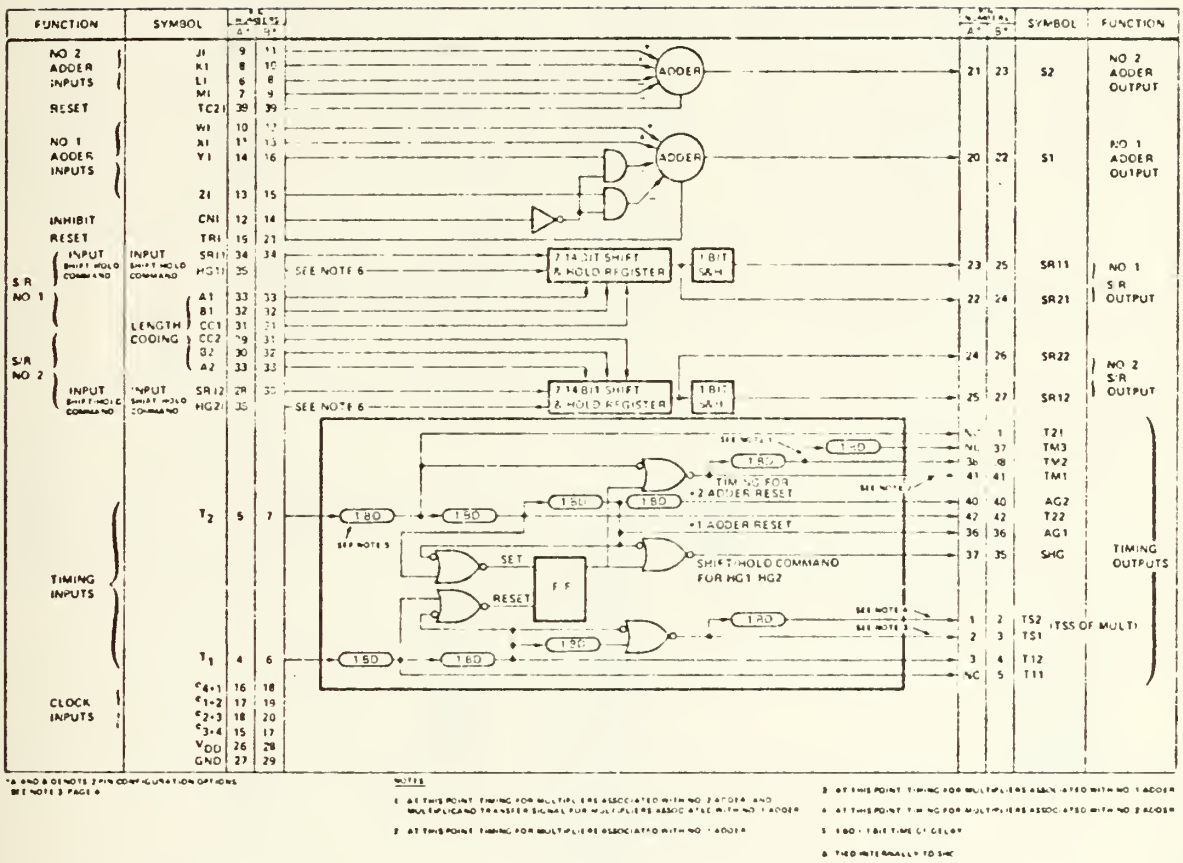


Figure 3-4. Block Logic Diagram of 65007NA/B Shift-Register/Adder

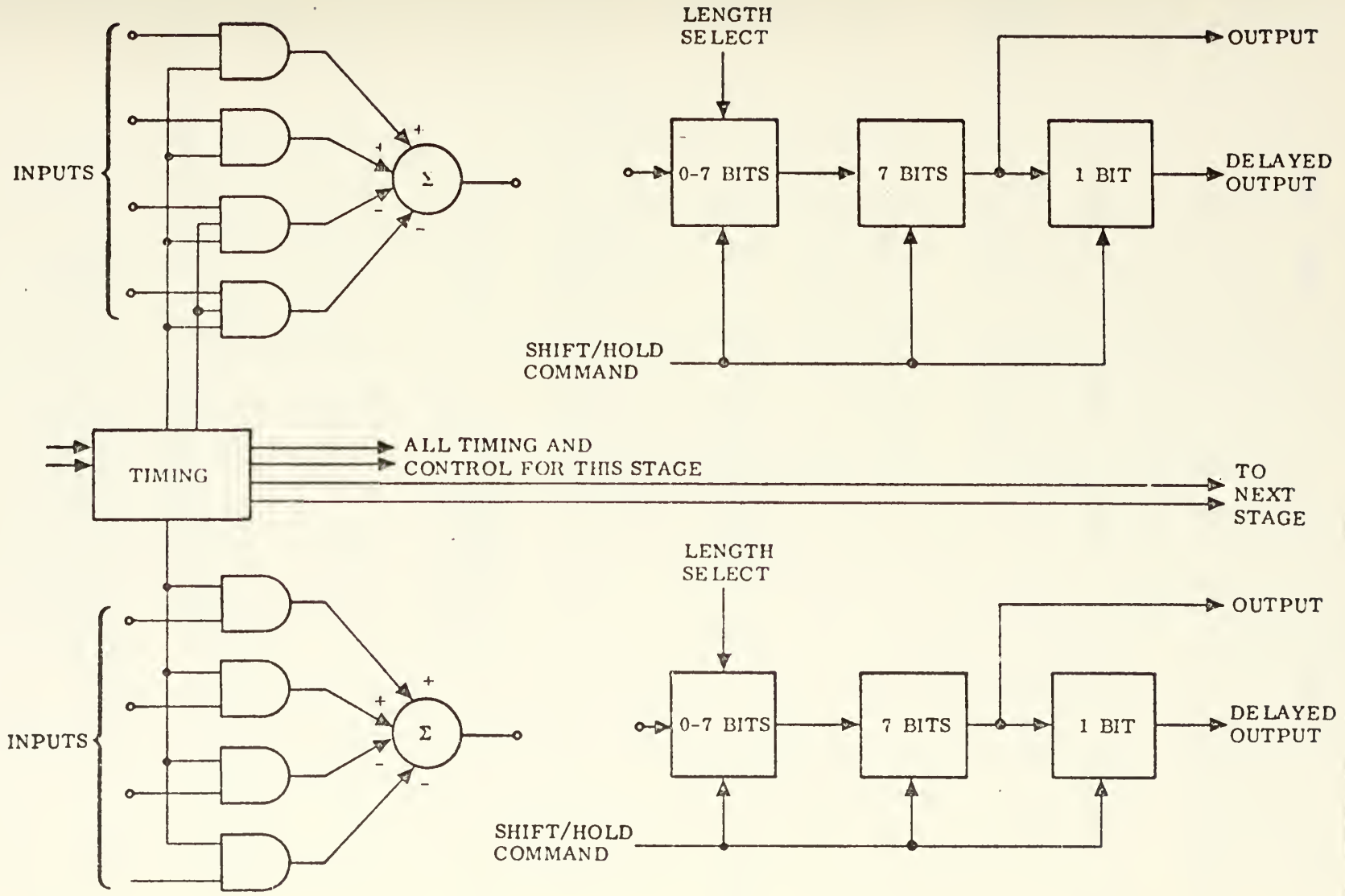


Figure 3-5. Simplified Logic Organization of Shift Register/Adder

The canonic form is the one generally used to realize second order sections, since minimizing the number of operations (particularly multiplications) corresponds to a minimum number of noise error sources due to quantization (round-off or truncation) within the D.F.

P. Girard [25] extending a previous work by Parker and Hess [2], has shown that from the state equations and associated transfer functions

$$\underline{x}(n) = \underline{A}\,\underline{x}(n-1) + \underline{B}\,u(n-1)$$
$$v(n) = \underline{C}\,\underline{x}(n-1) + d\,u(n-1)$$

$$H_S(z) = d + \frac{c\,z^{-1} + e\,z^{-2}}{1 + a\,z^{-1} + b\,z^{-2}} \tag{3.9}$$

$$\underline{x}(n) = \underline{A}\,\underline{x}(n-1) + \underline{B}'\,u(n-1)$$
$$v(n) = \underline{C}'\,\underline{x}(n) + d'\,u(n-1)$$

$$H_T(z) = d' + \frac{c' + e'\,z^{-1}}{1 + a\,z^{-1} + b\,z^{-2}} \tag{3.10}$$

there are 36 canonic realizations for d = 1, 36 for d = 0, 22 for d' = 1 and 22 for d' = 0.

The most general form of the transfer function of a second order filter can be expressed as

$$H(z) = \frac{V(z)}{U(z)} = K\,\frac{1 + a_1\,z^{-1} + b_1\,z^{-2}}{1 + a\,z^{-1} + b\,z^{-2}} \tag{3.11}$$

from which eq. (3.9) can be obtained by dividing the denominator into the numerator in ascending powers of z^{-1}. Equation (3.10) can also be obtained from eq. (3.11), if $b \neq 0$, by dividing the denominator into the numerator in descending powers of z^{-1}.

Only poles and zeros within the unit circle (in the z plane) will be considered, since it corresponds to minimum phase stable filters. Therefore the magnitude of the coefficients "b_1" and "b" are less than unity and the magnitude of the coefficients "a_1" and "a" are less than two.

Equation (3.11) is easily mechanized in the S_{a_3} form [2], also called SM_{11} form [25], as shown in Figure 3-6. z^{-1} is the unity delay operator and the multiplier gains are the coefficients K, a_1b_1, a and b.

M_0 sets the scaling coefficient (K)

M_1 sets a/2, which affects the resonant frequency of the pole.

M_2 sets b, which affects the damping of the pole.

M_3 sets $a_1/2$, which affects the frequency of the zero.

M_4 sets b_1, which affects the depth of notch of the zeros.

Since a and a_1 can be as large as two, the multipliers M_1 and M_3 are set at half value but summed twice at the adders. This will assure that the multipliers will perform the proper sign connection since all inputs will be less than unity.

This configuration is capable of realizing real and complex pairs of poles and zeros within the unit circle.

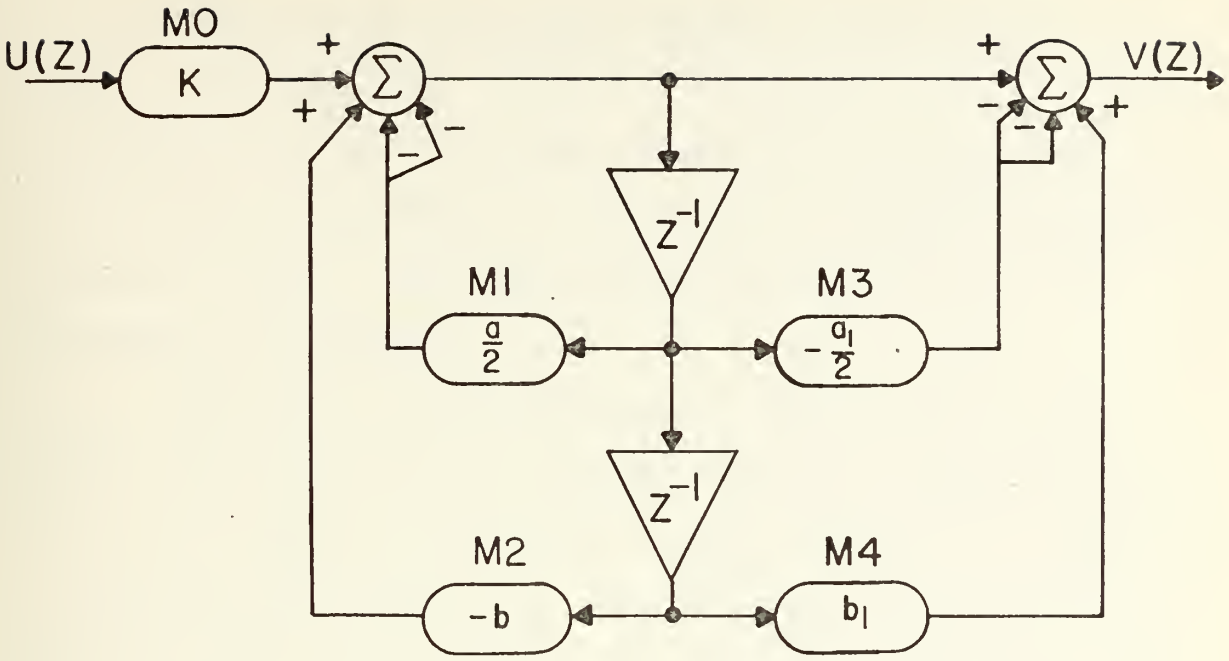


FIGURE 3-6 RECURSIVE CANONICAL REALIZATION OF A SECOND ORDER FILTER SECTION ON SM_{II} FORM

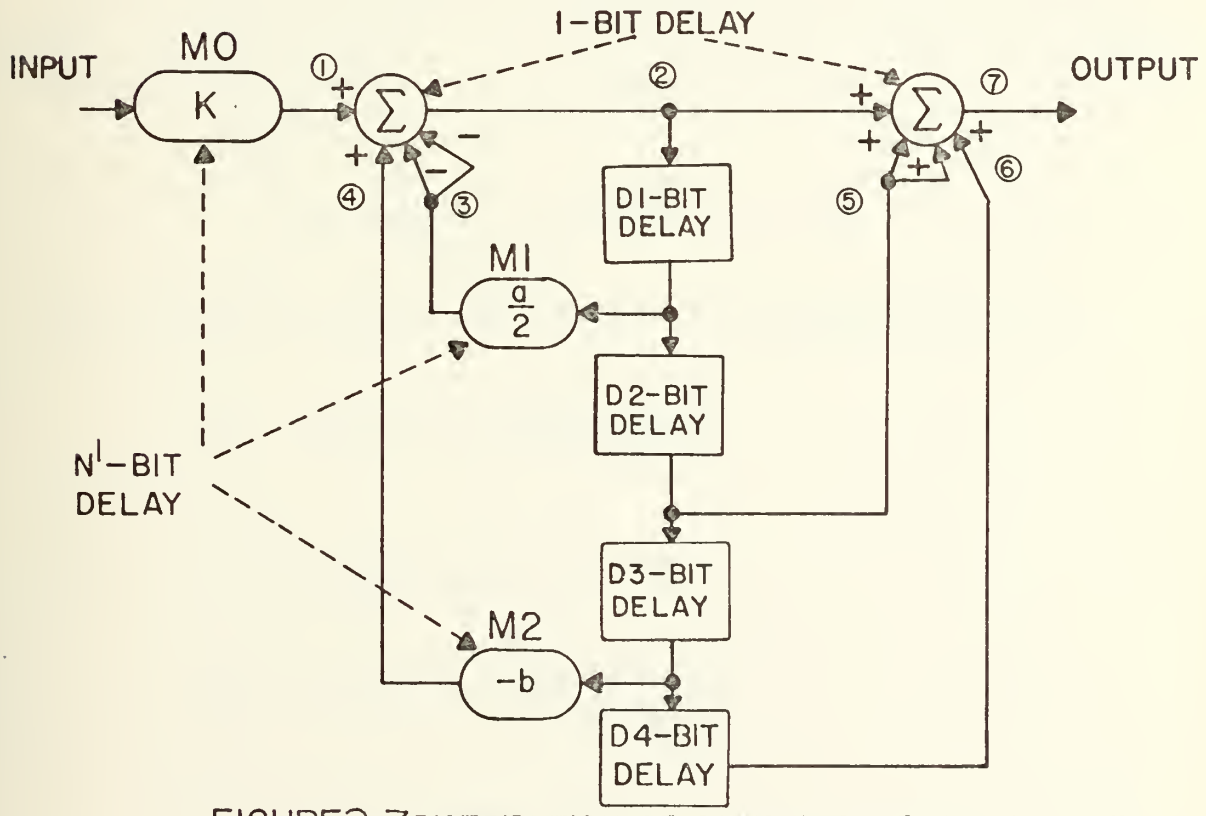


FIGURE 3-7 DISTRIBUTION OF GAINS AND DELAYS ON THE SECOND ORDER LOW PASS FILTER EXAMPLE

3. Example of a Low Pass Digital Filter Design

Assuming that a digital filter for a 10 KHz rate is required such that it is flat to 3 dB in the passband of 0 to 1,000 Hz and which is more than 10 dB down at frequencies beyond 2,000 Hz. The filter must also be monotonic in passband and stopband.

Observing that a Butterworth filter can meet the above requirements in the analog domain and taking advantage of the knowledge of the analog design, the use of a transform technique seems convenient. The bilinear transform will be used, because it is the most applicable for constant magnitude passband and stopband, as mentioned in Appendix B. But since the bilinear z-transform distorts the frequency response, a counter warp will be used on the design of the analog filter substituting each critical frequency ω_i by $(2/T) \tan (\omega_i T/2)$.

Since

$$T = 1/f_w = 1/(10 \text{ KHz})$$

then, each counter warped critical frequency will be

$$\omega_1' = (2/T) \tan \frac{(2\pi)\ (1 \text{ KHz})}{2\ (10 \text{ KHz})}$$

$$= (2/T)(.3249)$$

$$\omega_2' = (2/T) \tan \frac{(2\pi)\ (2 \text{ KHz})}{2\ (10 \text{ KHz})}$$

$$= (2/T)(.7265)$$

The cut off frequency is specified by the 3 dB point; then, in this case

$$\omega_c' = \omega_1' = (2/T)(.3249)$$

Applying the Butterworth analog design method

$$(Vp/V)^2 = 1 + (x/x_{3dB})^{2n}$$

where for a low pass filter $x = \omega$ and $x_{3dB} = \omega_c$ and V_p is the peak amplitude

V is the amplitude at a given point x

n is the order of the filter

Since $V_p/V_2 = 10$ dB then $(V_p/V_2)^2 = 10$ and the order of the filter can be obtained from

$$1 + \left(\frac{\omega_2}{\omega_c}\right)^{2n} \geq 10 \qquad \text{giving } n = 2$$

Then

$$H(\omega) = \frac{1}{1 + (\omega/\omega_c)^{2n}} = \frac{1}{1 + (\omega/\omega_c)^4}$$

and

$$H(s) = \frac{1}{(s/\omega_c)^2 + 1.414(s/\omega_c) + 1}$$

Replacing s by $(2/T)\frac{z-1}{z+1}$

and since $\omega_c = (2/T)(.3249)$, yields the required transfer function in the z-domain

$$H(z) = \frac{.0675569(z^2 + 2z + 1)}{z^2 - 1.14216\ z + .41244}$$

which can be written in the form of equation (3.11)

$$H'(z) = K\ \frac{1 + 2\ z^{-1} + z^{-2}}{1 - 1.14216\ z^{-2} + .41244\ z^{-2}}$$

where

$$a = -1.14216$$

$$b = .41244$$

$$a_1 = 2$$

$$b_1 = 1$$

and K is the scaling factor necessary to avoid overflow.

$$K \leq \frac{|\text{Denominator}|\ \min}{|\text{Numerator}\ |\ \max} = \frac{1 - |a| + b}{1 + |a_1| + b_1}$$

$$= \frac{1 - |1.1422| + .4124}{1 + |2| + 1} = .06755$$

Using the mechanization shown in Figure 3-6 it can be observed that with the multiplier coefficients previously calculated, the multipliers M3 and M4 are not necessary. Therefore a realization of the type presented in Figure 3-7 will be attempted. The timing distribution calculation will give the required delays (D_1,D_2,D_3 and D_4) to the shift registers.

Assuming the same accuracy in all multiplier coefficients, each multiplier will present N' - bit delay and each adder 1 - bit delay. For a computational word length M', a restriction is given by equation (3.8). From this equation since the chips can not operate at a bit rate higher than 1 MHz and a sampling rate of 10 KHz is required, then the word time M' + N' must be less than 100.

Since the data at (3) must be in word synchronization with (1), but delayed one word time

$$1 + D1 + N' = M' + N' \qquad \text{then} \quad D1 = M' - 1$$

and similarly with the data at (2) and (5)

$$D1 + D2 - M' + N' \qquad \text{then} \quad D2 = N' + 1$$

The data at (4) has to be delayed two word times from the data at (1) and in word synchronization with it

$$D1 + D2 + D3 + N' = 2(M' + N') \quad \text{then } D3 = M' - 1$$

Finally, comparing the data at (6) with the data at (5) we can obtain

$$D3 + D4 = M' + N' \qquad \text{then} \quad D4 = N' + 1$$

For a precision of 5 decimals on the coefficients of the multipliers, the use of equation (2.1) will indicate the need of 17 bits. One SPM chip will permit only a coefficient up to 8-bit-plus sign. Two SPM chips will permit up to 16-bit plus sign (N' = 17 bits). Each multiplication will be realized cascading two SPM, and therefore six SPM chips will be required.

The computational word length (M') has to be larger than the word length out of the A/D converter and should be made large enough to compensate for truncation errors in the filter computation. Choosing M' = 30 bits and recalling that each SRA chip provides two separate shift registers capable of delaying up to 15 bits, it can be concluded from the timing calculations made previously that four SRA chips are required, since D1, D2, D3 and D4 need 29, 18, 29 and 18-bit delays, respectively.

However, a better solution can be achieved using only two SRA chips and an extra multiplier (M3). This multiplier is set with a fixed coefficient of minus one in order to permit two additions and two subtractions at the output of the SRA, as shown in Figure 3-8. Therefore, D2 = N' + 1 bit delays are obtained with N' - bit of the

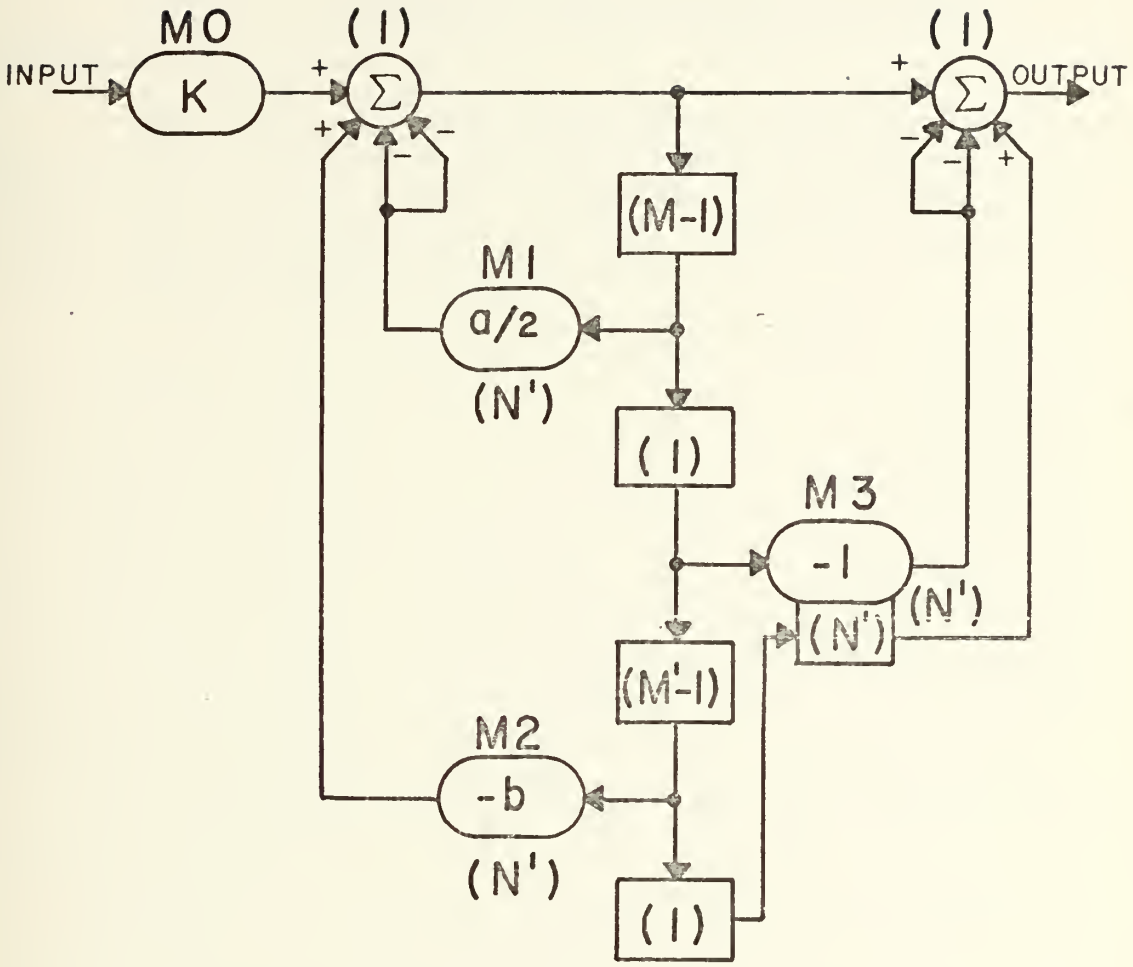


FIGURE 3-8 BLOCK DIAGRAM OF A SECOND ORDER LOW PASS FILTER IMPLEMENTATION SHOWING TIMING DISTRIBUTION.

multiplication process plus one bit delay available from the previous shift register, which uses M' - 1 bit delay. In order to obtain D4 = N' + 1 bit delays, the shift register of the multiplier M3 is used giving N' bit delays and as before one bit is available from the previous shift register (D3 = M' - 1).

For the chosen word lengths M' = 30 bits and N' = 17 bits, only four SPM and two SRA chips will be required, rather than three SPM and four SRA.

IV. DESIGN OF A SECOND ORDER DIGITAL FILTER SECTION USING THE SM_{11} TRANSPOSE FORM

A. INTRODUCTION

A second order building section in the SM_{11}^{T} form (transpose of SM_{11}) has been designed able to perform with the digital filter laboratory unit built by S.A. White from the North American Rockwell Electronics Group.

In order to permit the same parameter variations, the designed section is capable of a computational word length (M') from 16 to 30 bits and multiplier coefficient (N') 12, 14 or 17. The length of both these words as mentioned previously, affect the accuracy and the speed of the digital filter. The clock frequency is variable between 25 KHz and 1 MHz. The filter sampling rate is related to the previous variables by the equation (3.8).

The second order building block implements the following expression

$$Y(z) = K \frac{1 + a_1 z^{-1} + b_1 z^{-2}}{1 + a z^{-1} + b z^{-2}} X_1(z) + X_2(z) - X_3(z) - X_4(z) + X_5(z) - X_6(z) - X_7(z) \qquad (4.1)$$

The following state equation

$$\underline{x}(n) = \underline{A}\,\underline{x}(n-1) + \underline{B}\,u(n-1)$$

$$v(n) = \underline{C}\,\underline{x}(n-1) + d\,u(n-1)$$

for a single input single output second order filter leading to the S type transfer function indicated in equation (3.9) can be written in the form

$$\begin{bmatrix} x_1(n) \\ x_2(n) \\ v(n) \end{bmatrix} = \begin{bmatrix} 3 \times 3 \\ \text{array} \end{bmatrix} \begin{bmatrix} x_1(n-1) \\ x_2(n-1) \\ u(n-1) \end{bmatrix} \qquad (4.2)$$

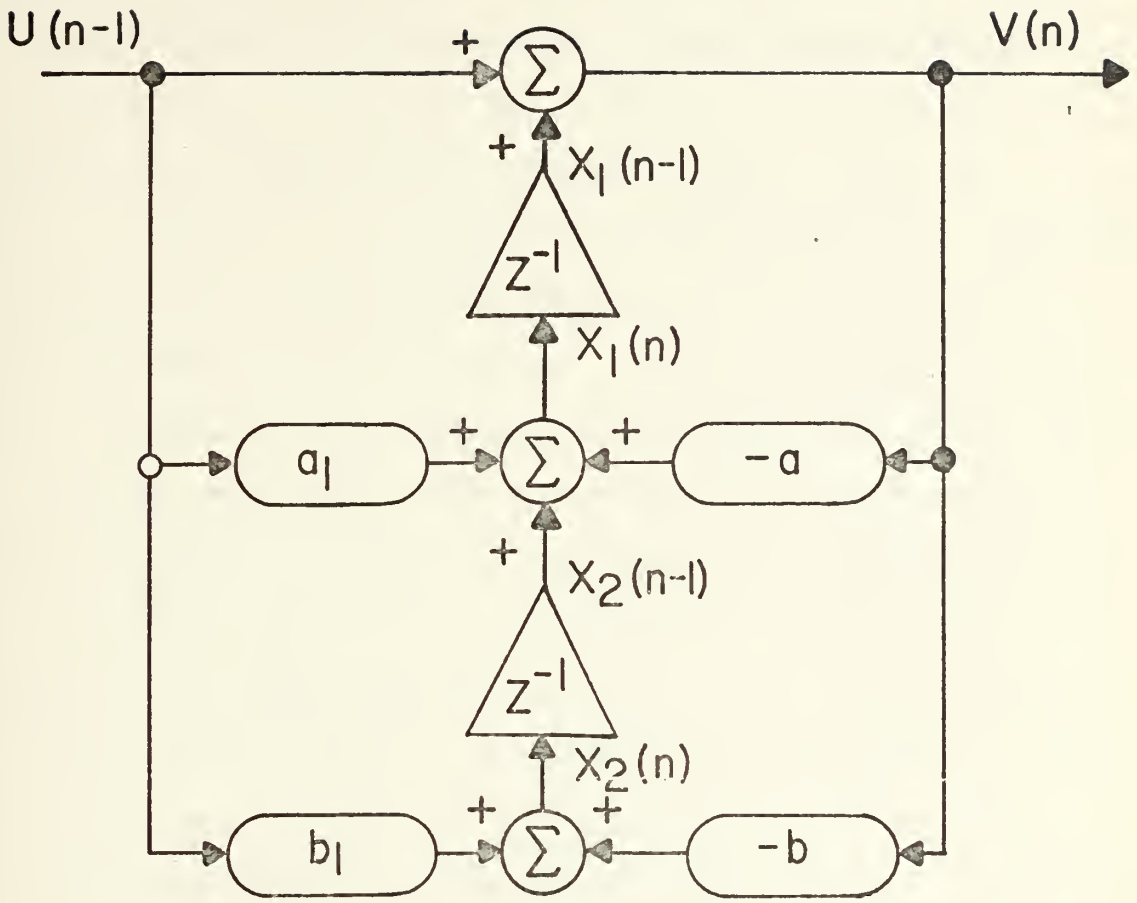
P. Girard [25] has introduced the canonical arrays, which corresponds to the idea of canonical realization given by Jackson [4].

The SM_{11} transpose array has the following form

$$SM_{11}^{T} = \begin{bmatrix} -a & 1 & a_1-a \\ -b & 0 & b_1-b \\ 1 & 0 & 1 \end{bmatrix} \qquad (4.3)$$

This is a canonical array since its realization minimizes the number of operations required, therefore leading to smaller quantization errors. This realization satisfied equation (4.2) for the canonical array (4.3) and the defined state vector $\underline{x}(n)$, as shown in Figure 4-1.

The coefficients a_1 and b_1 are related with the ones of the transfer function (3.9): $a_1 = a + c$ and $b_1 = b + e$. For this realization $d = 1$.



$$\left[SM_{II}\right]^{T} = \begin{bmatrix} -a & 1 & a_1 - a \\ -b & 0 & b_1 - b \\ 1 & 0 & 1 \end{bmatrix}$$

FIGURE 4-1 CANONIC REALIZATION OF A SECOND ORDER SECTION BASED UPON THE SM_{II} TRANSPOSE ARRAY

B. STRUCTURE MECHANIZATION

The design will be restricted to stable minimum phase filters. Stability implies poles within the unit circle in the Z-plane or in a parameter plane $|a| < 2$ and $|b| < 1$. Minimum phase implies zeros within the unit circle or $|a_1| < 2$ and $|b_1| < 1$. Since for proper multiplier operation, the magnitude of the coefficient has to be less than one, some arrangement has to be made. In the multipliers M2 and M4 the coefficient introduced will be respectively $a_1/2$ and $a/2$, but as observed in Figure 4-2 the second half of the adder number one, A1(2), will sum twice the output coming from the first half of the shift register, SR(1), which is delaying the resulting information not only from M_2 and M_4 but also from M_1 and M_3. Therefore the coefficient of this last multiplier will be set at $b_1/2$ and $b/2$, respectively.

The block diagram mechanization presented in Figure 4-2, minimizes the number of devices required to perform a SM_{11} transpose form realization for the required specifications.

The truncation processed in the D.F. is generally represented after each multiplication, however the NRMEC chips perform the truncation at the input of each adder. No problem will occur if the realization is of the SM_{11} form as shown previously by Figure 3-6. However in a transpose realization the scaling coefficient multiplier, MO, is cascade with other multipliers. The truncation could be simply realized with an AND-gate controlled by a signal composed by a string of ones M' bits long. The first half

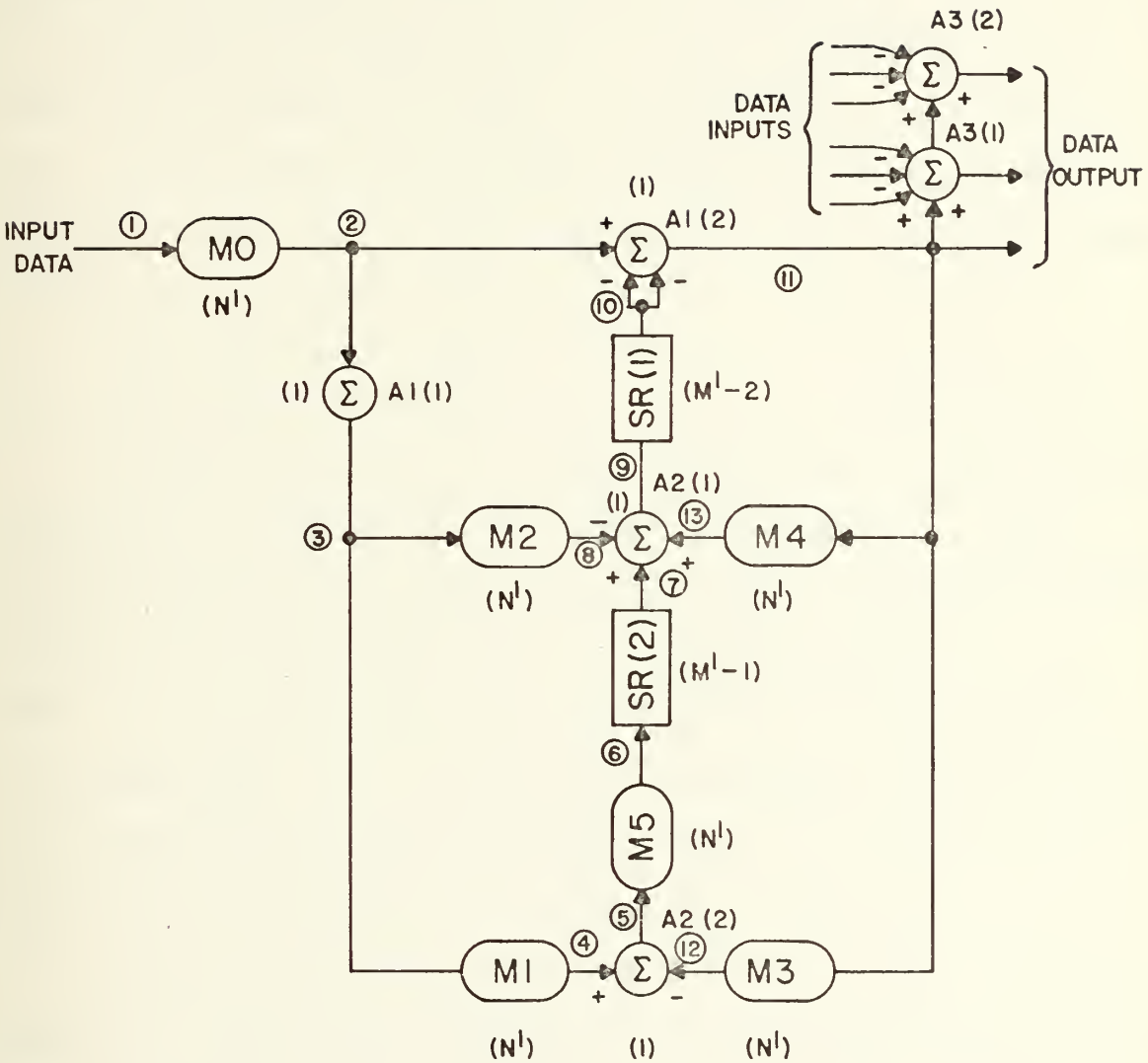


FIGURE 4-2 BLOC DIAGRAM OF A SECOND ORDER FILTER MECHANIZATION IN THE $[SM_{11}]^T$ FORM SHOWING TIMING DISTRIBUTION

of the adder number one, A1(1), has been utilized instead, since from the three SRA chips needed, only five adders were used. A1(1) also provides the necessary bit delay to obtain the synchronization of the signals (8) and (13) at the first half of the adder numbers two A2(1). The two adders of chip number three A3(1) and A3(2), facilitate the interconnection of other filter sections in parralel.

Multiplier M5 with a fixed scaling coefficient of -1, has been introduced in order to provide a N' - bit delay to the signal coming out of A2(2). Since the shift register of M5 is free, due to its fixed coefficient, it will be used to delay the synchronization signal N' - bit.

C. SHIFT REGISTER TIMING

The next step towards the implementation of this filter section is to determine the timing requirements. For a computational word length of M' bits and a multiplier coefficient of N' bits, correspond a multiplier output of (M' + N') bits, therefore a word time z^{-1} = (M' + N') bits is established. As before, each multiplier will be treated as presenting an effective delay of N' bit times, and that each adder will produce a one bit time delay.

The delay provided by the shift register SR(1) has to be such that the data at (10) are in word synchronization with, but delayed one word time from the data at (2) . Then, 1-bit delay at A1(1) plus N'-bit delay at M2 plus 1-bit delay at A2(1) plus the delay at SR(1) as to be equal to one word time, (M' + N') bits, or

$$1 + N' + 1 + \text{delay SR(1)} = M' + N'$$

then $$\text{delay SR(1)} = M' - 2$$

Similarly, the delay provided by SR(2) has to be such that the data at (7) are in word synchronization with, but delayed one word time from the data at (8). Starting from the signal at (3) :

$$N' + 1 + N' + \text{delay SR(2)} = N' + (M' + N')$$

then $$\text{delay SR(2)} = M' - 1$$

Since the computational word length M' can be as large as 30 bits, one entire SRA chip or two halves are required for each delay SR(1) and SR(2).

Next, it is necessary to verify that the data (4) and (12) entering A2(2) are in word synchronization. In fact starting from (2) , via M1, a delay of 1 + N' is obtained at (4) and via M3 the same delay is obtained at (12).

From Figure 4-2, it can be observed that the output presents a delay of (N' + 1) bits with respect to the input. Thus, for a synchronization input signal T_1, the corresponding synchro output is $T_1\, d^{N'+1}$, where d represents one bit delay time.

Figure 4-3 presents the wiring diagram of this filter section. The small numbers inside each box represent the pin number of the MOS chips. The multipliers are used in

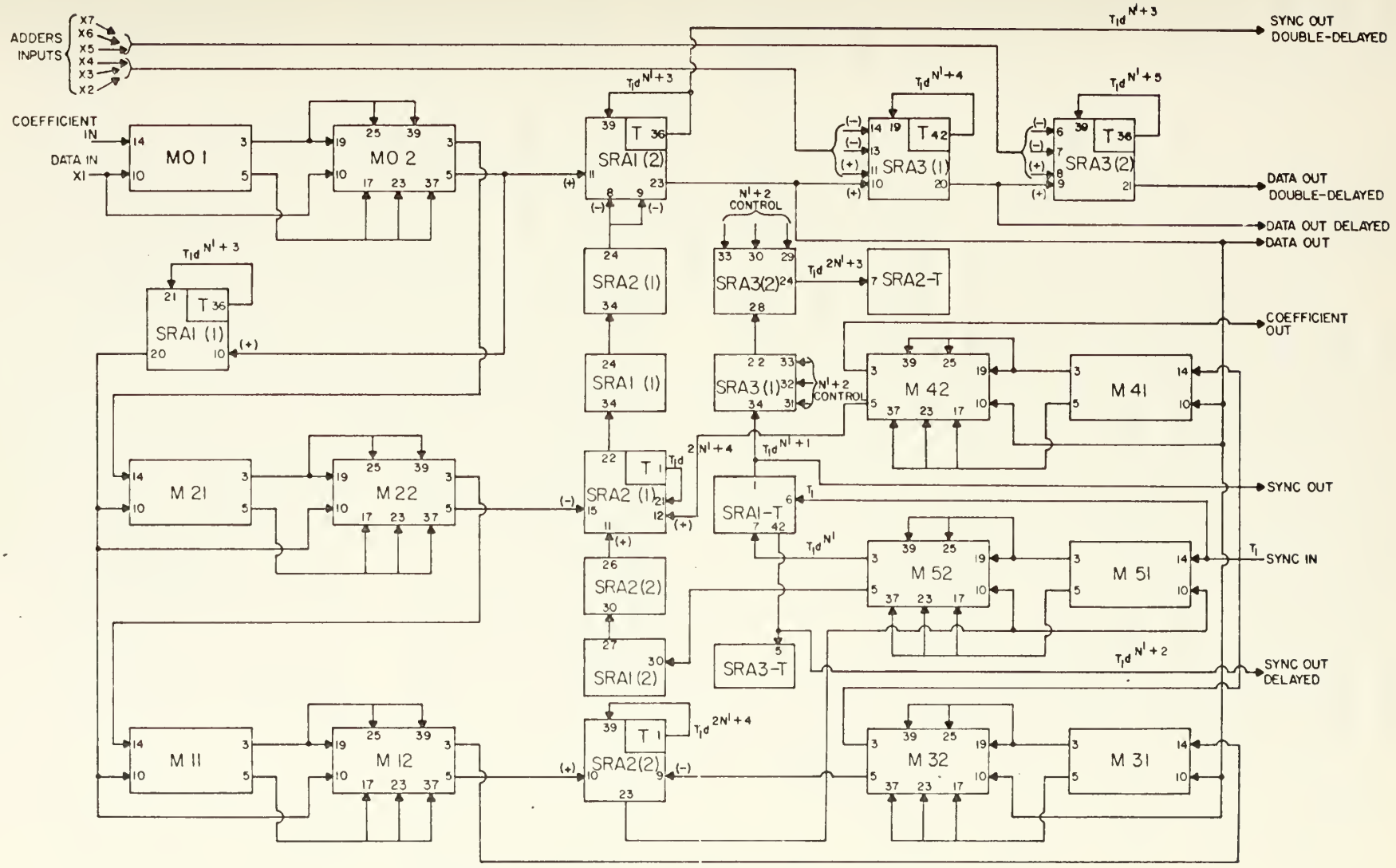


FIGURE 4-3 ASSEMBLY WIRING DIAGRAM OF A SECOND ORDER FILTER SECTION IMPLEMENTED IN THE SM_{II}^{T} FORM WITH NRMELC BUILDING CHIPS.

pairs to obtain the required coefficient accuracy (N' up to 17-bits). Then M0 becomes M01 and M02, etc. All multiplier shift registers are wired in series for serial loading of the multiplier coefficients. The scaling coefficient word is read into this shift register cyclically. The box marked T in the shift register adders represents the timing sections of these devices. Each T-section provides the proper timing signals not only to the proper SRA but also to the associated multipliers. SRA1 - T receives as inputs the signals T_1 and $T_1 d^{N'}$ and since an output with 1-bit delay is required, $T_1 d^{N'+1}$, SRA1 with type B pin configuration has to be used. For similar reasons, SRA2 will also be type B.

D. TIMING DIAGRAM

In order to illustrate the processing of the signal through the filter and obtain a timing diagram, the maximum word lengths for the computational loop (M' = 30) and for the multiplier coefficients (N' = 17) will be assumed and without loss of generality an input data signal of 15-bit plus sign will be considered.

The timing at the points marked with circled numbers in Figure 4-2 is illustrated in Figure 4-4. The data enters the scaling multiplier M0, at (1) at word time I, with the LSB input first and the sign bit 16-bits later. This data is represented shaded so that the propagation of that word through the filter can be traced by following the shaded data.

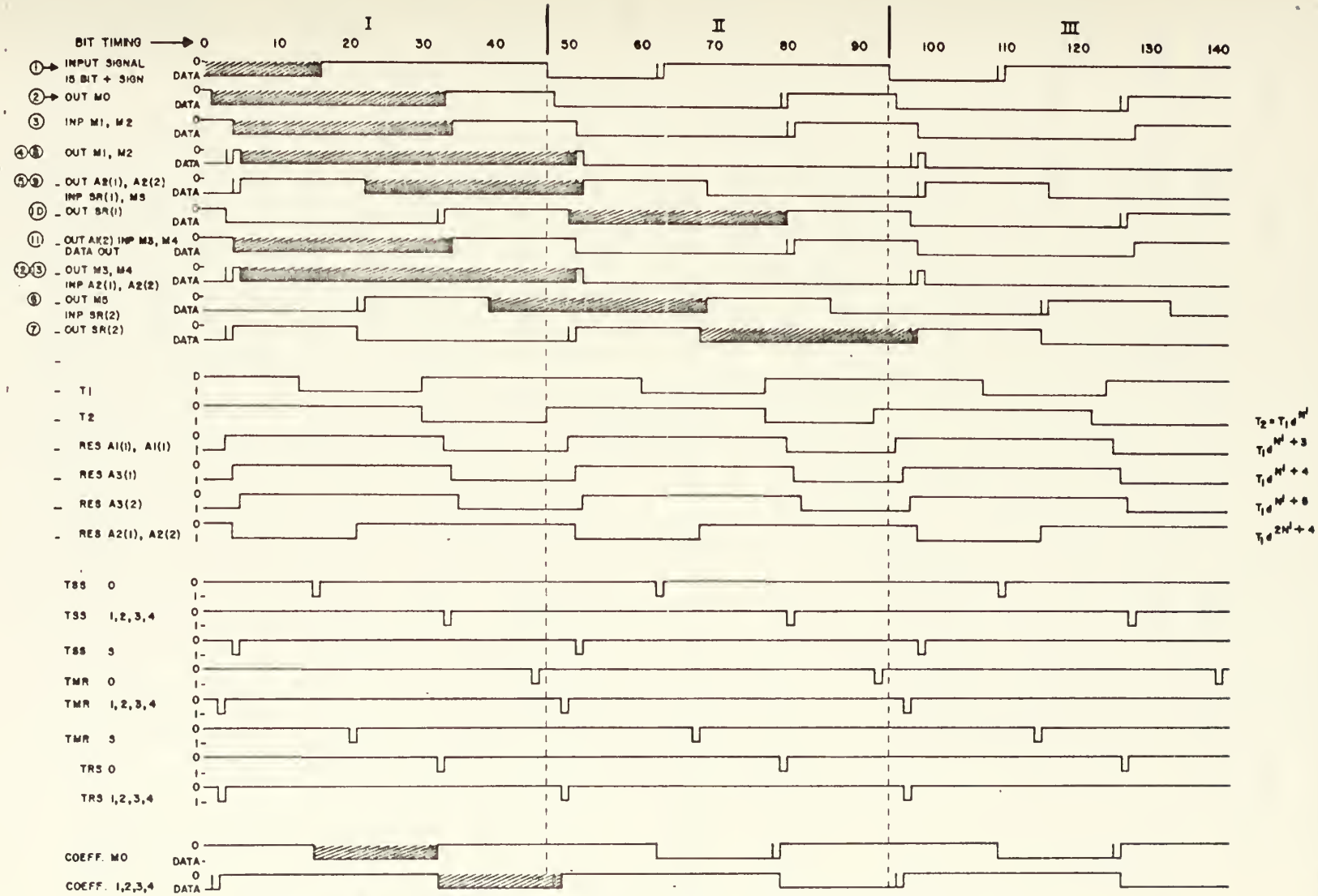


FIGURE 4-4 TIMING DIAGRAM FOR AN INPUT SIGNAL 15-BIT-PLUS-SIGN ($C^1=16$), SCALING COEFFICIENT 16-BIT-PLUS-SIGN ($N^1=17$) AND DATA COMPUTATIONAL WORDLENGTH 29-BIT-PLUS-SIGN ($M^1=30$).

The data at (2) are represented by a longer data word than the one at (1) because the multiplier generates a double-precision product (15 + 16 + 1 = 32-bit) and delayed N' = 17 bit. Then the data out of MO is longer than the computational word length. The truncation to 30 - bit will occur at the input of the adder, A1(1). The reset signal for this adder, shown at the line RES A1(1) of Figure 4-4, is off only for 30-bit, eliminating the two first bits being inputed to the adder. The data through A1(1) will be delayed 1-bit and as indicated at (3) will be 30-bit long, inputing the multiplier M1 and M2.

The data at (4) and (8), after weighted by the multipliers M1 and M2 respectively, will be $M + N + 1 = M' + N' - 1 = 30 + 17 - 1 = 46$ bit long and delayed N' = 17-bit from the multiplier input data at (3).

The data at (4) must be in word synchronization with the data at (12) inputing A2(2). The required truncation to 30-bit is operated at the input adder. The reset signal, RES A2(2), has to be T_1 delayed 38 bits or in general $T_1 d^{2N' + 4}$. The data at the output of this adder (5) is then 30 - bit long and 1 - bit delayed from its inputs.

The data at (6), due to the multiplication process will be again 46-bit long and delayed 17-bit from the input at (5). The shift register SR(2), implemented with the second half of the SRA's numbers one and two, as shown in Figure 4-3, will delay the data (6) by M' - 1 = 29 - bit.

The data at (7) must be in word synchronization with, and delayed one word time from the data at (8) and (13), inputing A2(1). The truncation of these data inputs are truncated to the computational word length at the input of this adder. A reset signal $T_1 d^{2N'+4} = T_1 d^{38}$ is required.

The data at (9) passes through the shift register SR(1) so that its output at (10) will be M' - 2 = 28 - bit delayed from (9).

The data at (10) has to be in word synchronization with the data at (2), inputing A1(2). Here, the truncation will affect only the data (2) since the data (10) resulting from delaying the output of an adder conserves the computational word length.

The data output of this filter section at (11) can be added with six more data inputs provided fron other filter sections for a parallel realization or cascaded with identical sections for a series realization.

E. DESIGN OF A SHIFT REGISTER CONTROLLED BY THE COEFFICIENT WORD LENGTH

As seen previously a reset signal delaying T_1 by (2N' + 4) bit is required for both adders A2(1) and A2(2). Since all multipliers are capable of control N', the shift register part of M5, can be used, because its coefficient (minus one) is fixed. In Figure 5-3 the output pin 3 of M52 provides a signal T_1 delayed N' - bit. Unfortunately, no other multiplier shift register is available to obtain a shift register controllable by the coefficient word length.

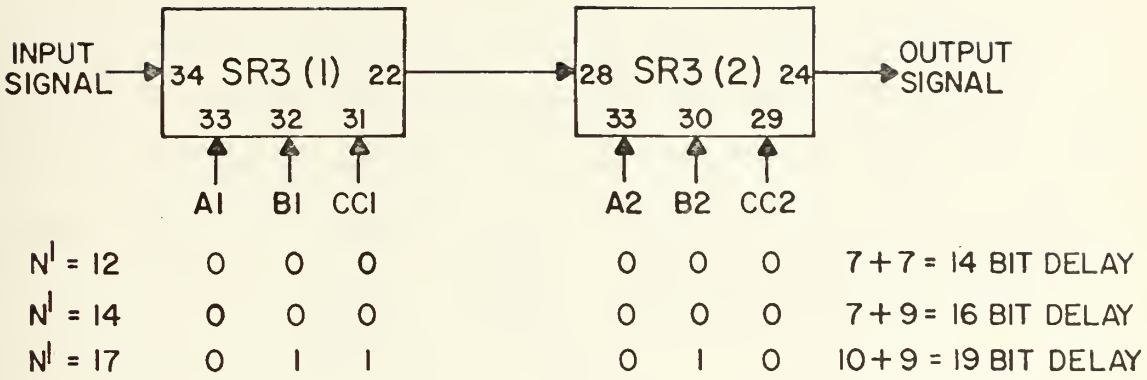
Figure 4-5a shows the wiring connections to a third shift register which can delay a signal by (N' + 2) bit delay. Figure 4-5b presents the design of a diode matrix able to control the length coding of that shift register.

The coefficient word length (N') can have the values 12, 14 and 17. If 12-bit has been chosen all shift register input length coding will be zeros, and a 7-bit delay is obtained at each one, resulting in an output 14 bit delayed. If a 14-bit coefficient word length is chosen, the multiplier selector switch set at 14, will put "1" on line B2, all other inputs remaining "0"'s and then SR(2) will produce 9 bit delay resulting in an output 16-bit delayed. If the multiplier selector switch is set at 17, lines B1, CC1 and B2 will go "1", then a 10-bit delay will be produced SR3(1) and a 9-bit delay at SR3(2), resulting in an output 19-bit delayed.

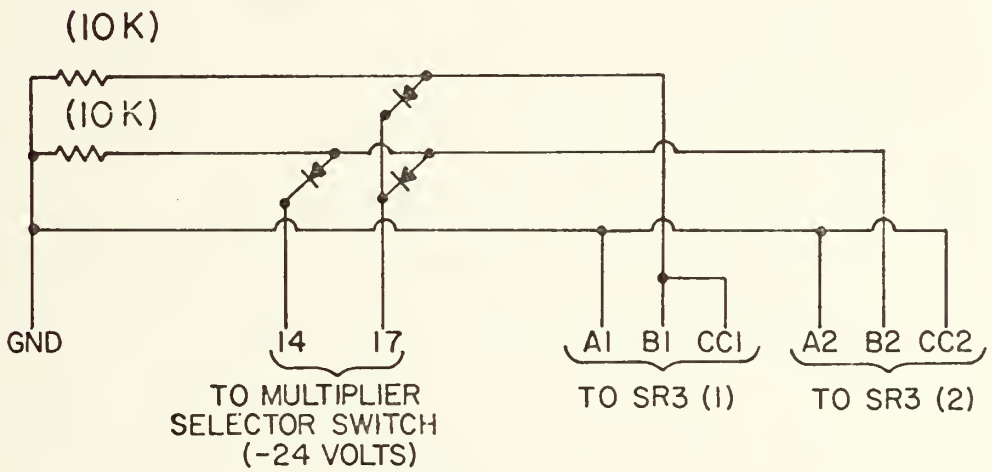
The shift register used (SRA 3) is package type A, since type B having a different pins connection, will not permit the proper code combination.

F. MULTIPLIER TIMING SIGNALS

The sign bit timing, TSS, is a one bit signal which goes "1" at the same time as the sign bit of the data appears at the multiplier serial input. Then, for the multiplier MO, TSS 0 appears at the 16th bit time as the sign bit at (1), and cyclically one word length (M' + N' = 47-bit)



(a)



(b)

FIGURE 4-5

(a) SHIFT REGISTER (TYPE A) CONNECTION TO OBTAIN ($N^1 + 2$) BIT DELAY

(b) COEFFICIENT WORD LENGTH DIODE MATRIX

later. Similarly the signals TSS 1,2,3,4 for the multiplier M1, M2, M3 and M4 and the signal TSS 5 for the multiplier M5.

The timing signal TMR is a one bit signal which goes "1" two bit time before the LSB of the data appears at the multiplier serial input. The multiplication starts at that time. Then, TMR 0 appears at the 46th bit time, two bits before the LSB appears at (1). Similarly, TMR 1,2,3,4 for M1, M2, M3, M4 and TMR 5 for M5.

The multiplicand transfer signal, TRS, transfers the serial multiplier coefficient input to a larallel register, after the whole signal be inputed. Then TRS goes "1" for one bit, one bit after the sign bit of the multiplier coefficient be inputed. Then, TRS 0 appears at the 33th bit time, one bit later than the sign bit of COEFF M0. Similarly, TRS 1,2,3,4 with respect to COEFF M1,2,3,4. The multiplier M5 does not need the TRS signal since its coefficient (-1) is fixed.

Although not represented in Figure 4-3, all data and synchronization filter outputs should have a buffer circuit to perform a convenient output isolation. The design of this buffer circuits and other controls however applicable to this design are not included, since they are referred to in [28].

V. QUANTIZATION AFTER ADDITION AND QUANTIZATION BEFORE MULTIPLICATION. ERROR BOUNDS

A. INTRODUCTION

When a digital filter is implemented, errors due to finite precision in the representation of the numbers always occurs. The word length after a multiplier or an adder is in general larger than the original word length. The case of increasing word length after an adder which results in "overflow" can be avoided by proper scaling at the input of the filter, as shown before. Therefore only the case of increasing word length after multiplication will be treated.

Up to now, the realization of D.F. has been done almost exclusively using special purpose computers. Thus in order to reduce storage, quantization is performed exactly when the number of bits is increasing, such as after multiplication. Almost all of the literature has been dedicated to the case of quantization after multiplication, either using a stochastic approach [5-6-20] or a deterministic one [1].

For hardware implementation of D.F.'s, for instance using the SRA (shift register adders) and SPM (serial parallel multiplier) chips from NRMEC, it is possible to maintain the resulting M' + N' - 1 bits after a multiplication of a N' bit multiplier times a M' bit multiplicand (sign bits included) until after next addition, because two consecutive multiplications will not occur (otherwise a single one would

suffice). It is possible to go even further, by carrying the M' + N' - 1 bits until the next multiplication will be performed. This leads to two new methods of performing the quantization. Namely, quantization after addition (QAA) and quantization before multiplication (QBM).

QAA only recently has been addressed [10-11], and shown that for the case of magnitude truncation, a second order D.F. has almost no limit cycles. QBM has not even been mentioned in the literature before.

It can be observed that for hardware implementation of D.F., using for instance NRMEC chips, the filter word length and the storage of the devices for the cases QAA and QBM are exactly the same as when used with QAM (quantization after multiplication). For this last case the adder would be active for M' bits (wordlength of the computational loop in the filter) and off for the remaining N' bits of filter wordlength (z^{-1} = M' + N' bits). However for QAA or QBM, the adder will be active for the M' + N' - 1 bits from the previous multiplication.

B. ADVANTAGES OF QAA AND QBM

It will be proved later that QAA will produce no larger quantization error bound than QAM, and that the error bound for QBM is smaller or equal to the QAA. In Appendix C, Lyapunov's direct method is applied to find the amplitude bound of the limit cycles in the second order D.F. assuming QAA. The result obtained is two times smaller than that

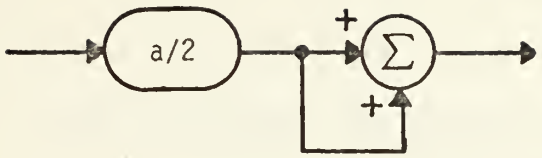
determined by Parker and Hess [1] for the case of QAM. Another advantage of using QAA or QBM over QAM is shown next.

In Chapter III it was mentioned that the magnitude of the multiplier coefficient has to be less than one in order to allow a proper operation of the SPM. From the examples presented in Chapter III and IV, it has been observed that whenever the magnitude of the multiplier coefficient is as large as two, as is common practice, one can introduce one half of the multiplier coefficient and sum twice the multiplier output at the next adder, as shown in Figure 5-1a.

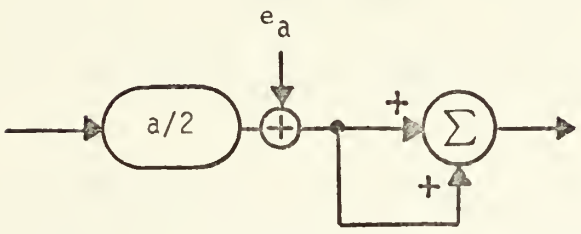
If finite arithmetic is now considered the output quantization errors for QAM and for QAA will be different. Consider, for instance, an input signal weighted by a coefficient ($|a| \leq 2$) and that rounding with a quantization step of h being used. For QAM, the maximum errors introduced after multiplication will be $|e_a| = h/2$ and, since the output of the multiplier is added twice at the adder as shown in Figure 5-1b, the maximum output errors will be h. For QAA, as shown in Figure 5-1c, the maximum magnitude output error will be h/2. Therefore two times smaller than for QBM.

C. HARDWARE MODIFICATIONS TO PERFORM QBM

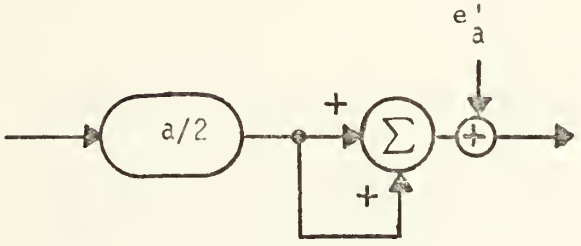
According to the reasons presented earlier, a hardware design able to perform QBM seems convenient. The NRMEC chips described in Chapter III could only perform truncation before each addition, which is equivalent to QAM (truncation)



(a)



(b)



(c)

Figure 5-1. Advantage of QAA over QAM When the Magnitude of the Coefficient Multiplier is Larger than One, Shown for $|a| \leq 2$.

if two multipliers are not cascade. However the NRMEC chips can easily be modified so that they are able to perform truncation or rounding before multiplication.

1. Serial/parallel Multiplier Performing Truncation or Rounding Before Multiplication

One way to obtain QBM, using truncation or rounding as desired, is to precede each SPM with a circuit as shown in Figure 5-2. It consists of one full adder and 2 flip-flop's acting as delay elements. An inverter is used in the carry circuit of the standard full adder integrated circuit.

Another way is to design a new SPM with the circuit described above included within the chips, as shown in Figure 5-3. Since the present SPM chip has 34 pads and it is mounted in a 42-lead pack, the three new inputs required (t, MI2 and r) can easily be placed in the available package pins.

The operation of the circuit presented in Figure 5-3 can be described as follows. Due to a previous multiplication the input to a multiplier can be as large as M' + N' - 1 bit, where M' and N' represent, respectively, the number of bits of the computational loop within the filter and the number of bits of the coefficient multiplier (sign bit included). At the beginning of the present multiplication this data input can not be larger than the computational word length (M'), in order that no more than M' + N' - 1

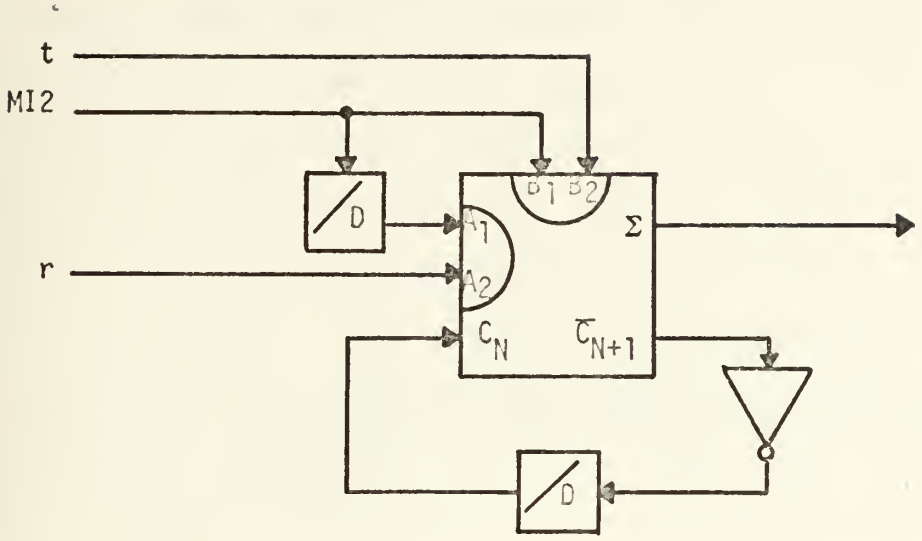


Figure 5-2. Two's Complement Truncation/Rounding Circuit

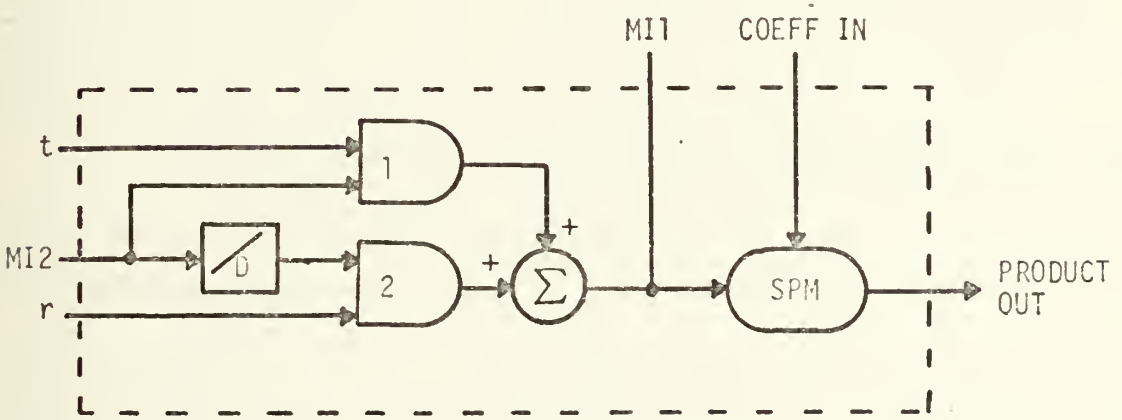


Figure 5-3. Modified SPM to Perform Truncation or Rounding Before Multiplication

bits appear at the output produced to avoid overlapping with the next word. Then truncation or rounding is required at the data input at MI2 to reduce this information to M' bits or, in other words, to eliminate up to N'-1 bits. If truncation is desired the input "r" is grounded and the input "t" will receive a signal as shown in Figure 5-4; a string of ones M' bit long starting M' bits prior to the sign bit be input at MI2. The output of AND gate number 2 will be always zero, and the AND gate number 1 will eliminate any information until "t" goes "1". Then, for a M' + N' - 1 bit input, the first N'-1 bits will be suppressed, and the information entering the SPM will have N' bits and will be 1-bit delayed by the adder.

If the input data already has M' bits or less, no bit will be eliminated using input MI2, but the 1-bit delay at the adder will exist. In order to eliminate the delay in this case, the input MI1 has been made available.

If rounding is required, both the signals "t" and "r" will be present. The rounding signal, "r', is a 1-bit signal which goes "1" M' bit prior to the sign bit of the data inputed at MI2. This signal will appear at the input of the gate 2 at the same time as the most significant bit (MSB) of the information being eliminated. This will be the only information passing gate 2. The output of gate 1, will truncate the input to M' bit as before, but now the previous MSB will be added to the LSB of the M' bit information. Thus a rounded M' bit data will input the SPM.

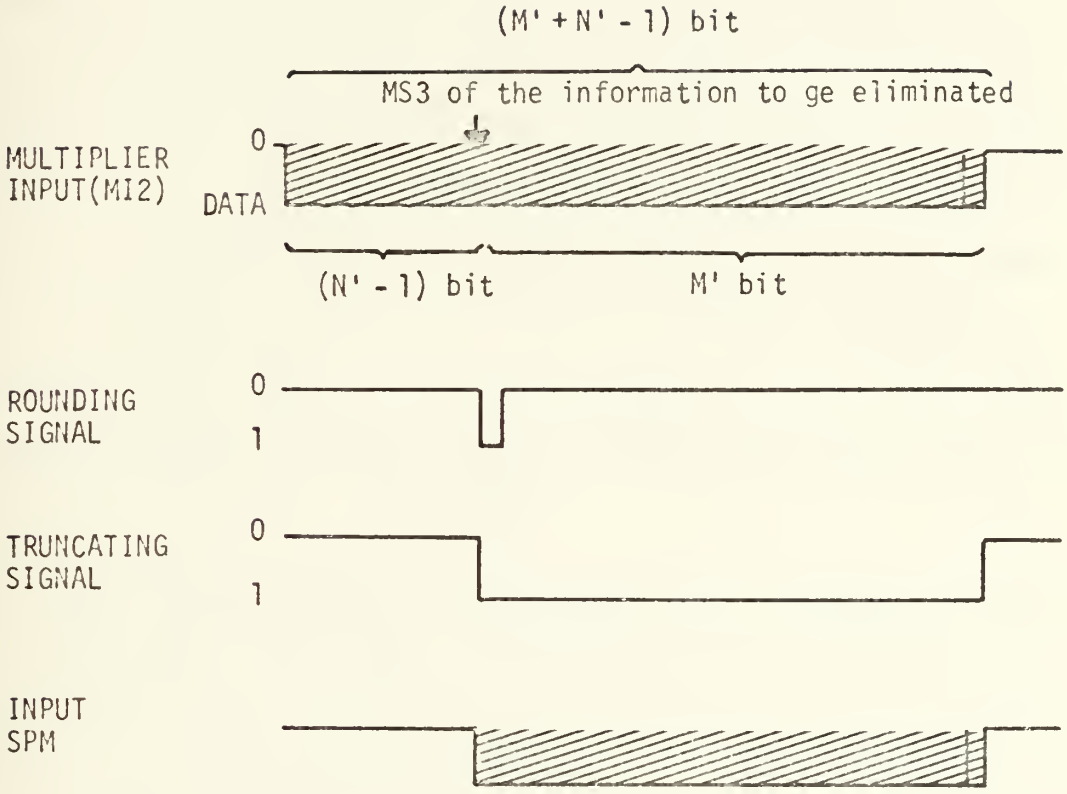


Figure 5-4. Timing Signals for the Modified SPM Shown in Figure 5-3

2. SRA Circuitry For Quantization Before Multiplication

The shift register adder chip itself requires no alteration for the QBM operation. Only the reset signal going to the adders must be modified. As shown in the timing diagrams, Figures 3-3 and 4-4, this reset signal was "0" during the M' bit prior to the input of the sign bit of the data being added, and "1" for the remaining N' bit time. Therefore the addition process was performed only during the last M' bit. For the QBM operation, the adder has to be active during (M' + N' - 1) bits. Then the reset signal has to be "0" during the (M' + N' - 1) bit information entering the adder or, in other words, it will be a 1-bit signal going to one the next bit after the sign bits of the data are inputed to the adder.

D. ERROR BOUNDS DUE TO FINITE PRECISION ARITHMETIC IN D.F.'S

Using the state space formulation of a second order digital filter, the difference between the states and outputs of a finite fixed point arithmetic D.F. and its infinite precision (ideal) counterpart is derived for the new quantization methods (QAA and QBM) introduced earlier. The QAA bound derivation follows a similar path used by S.R. Parker and Yakowitz [32] on their quantization after multiplication study. A different approach is required to compare QAA with QBM. Rounding is assumed with quantization step $\pm h/2$.

1. Quantization After Addition (QAA)

The state equations for an ideal (infinite precision) single-input single-output second order D.F., can be expressed as follows:

$$\begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1(n-1) \\ x_2(n-1) \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u(n-1) \tag{5.1}$$

$$v(n) = \begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{bmatrix} x_1(n-1) \\ x_2(n-1) \end{bmatrix} + d \quad u(n-1)$$

or in vector notation

$$\begin{aligned} \underline{x}(n) &= \underline{A}\ \underline{x}(n-1) + \underline{B}\ u(n-1) \\ v(n) &= \underline{C}\ \underline{x}(n-1) + d\ u(n-1) \end{aligned} \tag{5.2}$$

Assuming quantization after addition (QAA) for the finite precision D.F., as shown in Figure 5-5, the following state equations apply:

$$\begin{aligned} x^*_1(n) &= [a_{11}\ x^*_1(n-1) + a_{12}\ x^*_2(n-1) + b_1\ u(n-1)]_q \\ x^*_2(n) &= [a_{21}\ x^*_1(n-1) + a_{22}\ x^*_2(n-1) + b_2\ u(n-1)]_q \\ v^*(n) &= [c_1\ x^*_1(n-1) + c_2\ x^*_2(n-1) + d\ u(n-1)]_q \end{aligned} \tag{5.3}$$

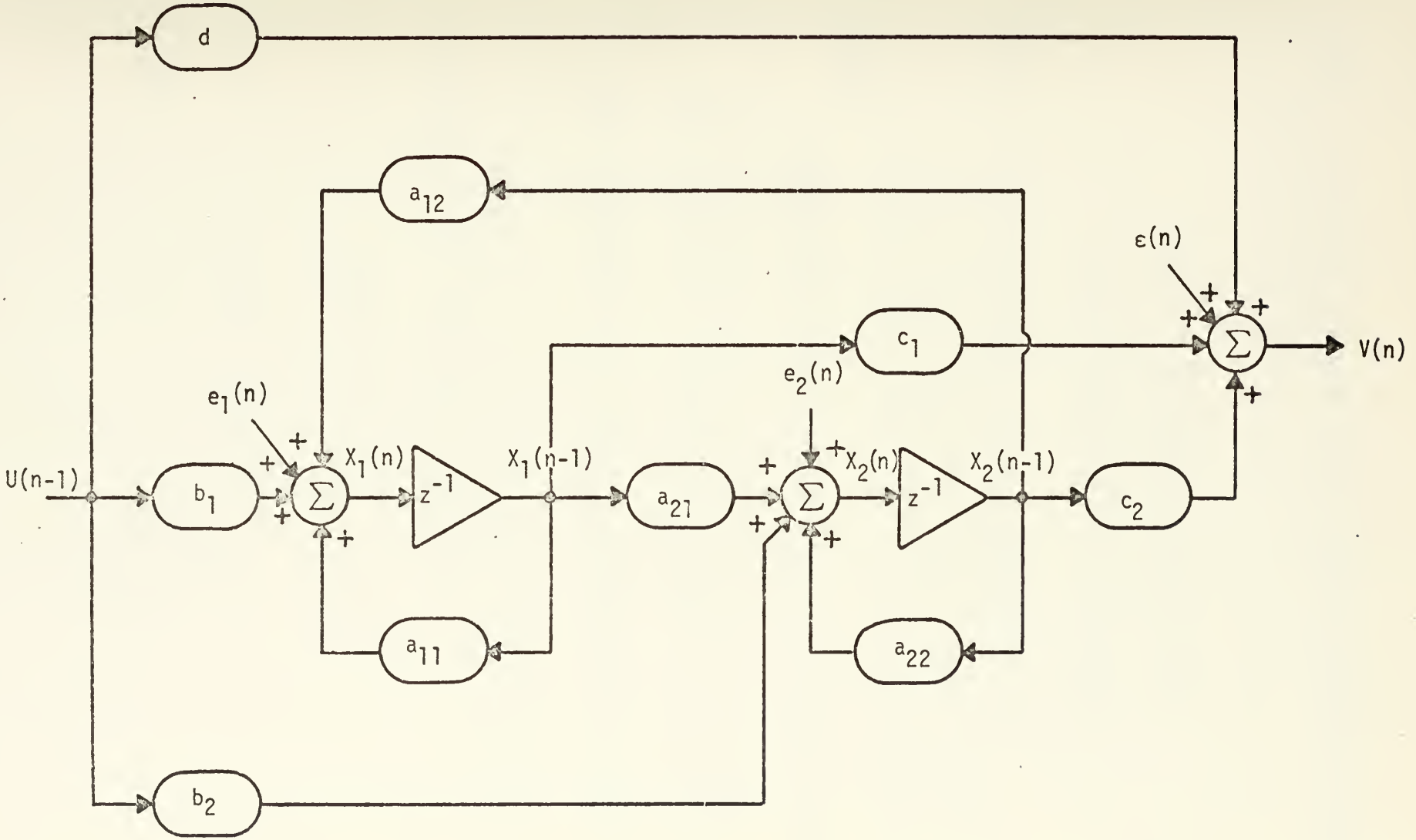


Figure 5-5. Second Order Single-Input Single-Output Digital Filter. Quantization After Addition

where * indicates signals in the finite precision filter. Or, in vector notation,

$$\begin{aligned} \underline{x}^*(n) &= [\underline{A}\ \underline{x}^*(n-1) + \underline{B}\ u(n-1)]_q \\ v^*(n) &= [\underline{C}\ \underline{x}^*(n-1) + d\ u(n-1)]_q \end{aligned} \tag{5.4}$$

where the input has been assumed quantized i.e., $u^*(n-1) = [u(n-1)]_q$. The output appears also quantized, $v^*(n)$, so that it can used as input to a next second order stage.

Define the error vectors $\underline{e}(n)$ and $\varepsilon(n)$, as follows:

$$\begin{aligned} \underline{e}(n) &= \underline{A}\ \underline{x}^*(n-1) + \underline{B}\ u(n-1) - [\underline{A}\ \underline{x}^*(n-1) + \underline{B}\ u(n-1)]_q \\ \varepsilon(n) &= \underline{C}\ \underline{x}^*(n-1) + d\ u(n-1) - [\underline{C}\ \underline{x}^*(n-1) + d\ u(n-1)]_q \end{aligned} \tag{5.5}$$

Assuming rounding with a quantization step of $\pm h/2$, the above error vectors are bounded

$$\begin{aligned} |e_k(n)| &\leq \rho_k\ h/2 \qquad k = 1,2 \\ |\varepsilon(n)| &\leq \rho_3\ h/2 \end{aligned} \tag{5.6}$$

where

$$\rho_k = \begin{cases} 0 & \text{if all elements in the Kth row of the D.F. array are 0 or 1} \\ 1 & \text{otherwise} \end{cases}$$

$$k = 1,2,3$$

Therefore, it is possible to find constant vectors $\underline{e}$ and ε, whose elements are larger than the magnitude of the correspondent elements of $\underline{e}(n)$ and $\varepsilon(n)$. Then

$$\begin{aligned} &<\underline{e}(n)> < \underline{e} \\ &<\varepsilon(n)> < \varepsilon \end{aligned} \tag{5.7}$$

Defining the state and output errors the same way as in [32] there results analogously

$$\begin{aligned} \underline{y}(n) &= \underline{x}(n) - x^*(n) \\ &= \underline{A}\,\underline{x}(n-1) + \underline{B}\,u(n-1) - [\underline{A}\,\underline{x}(n-1) + \underline{B}\,u(n-1)]_q \\ &\quad - \underline{A}\,\underline{x}^*(n-1) + \underline{A}\,\underline{x}^*(n-1) \\ &= \underline{A}\,\underline{y}(n-1) + \underline{e}(n) \end{aligned} \tag{5.8}$$

$$\begin{aligned} \Delta v(n) &= v(n) - v^*(n) \\ &= \underline{C}\,x^*(n-1) + d\,u(n-1) - [\underline{C}\,\underline{x}^*(n-1) + d\,u(n-1)]_q \\ &\quad - \underline{C}\,\underline{x}^*(n-1) + \underline{C}\,\underline{x}^*(n-1) \end{aligned}$$

$$\Delta v(n) = \underline{C}\ \underline{y}(n-1) + \varepsilon(n) \tag{5.9}$$

The error propagation equation (5.8) and the output error equation (5.9) have exactly the same form as the ones derived for rounding after multiplication in [32], and therefore lead to a state error magnitude vector

$$\langle \underline{y}(n) \rangle \leq \sum_{\ell=0}^{n} \langle \underline{A}^{\ell} \rangle\ \underline{e} \tag{5.10}$$

and an output error magnitude bound

$$\langle \Delta v(n) \rangle \leq \langle \underline{C} \rangle\ \langle \underline{y}(n-1) \rangle + \varepsilon \tag{5.11}$$

The bounds on the errors for QAA as indicated by (5.11) are at most as large as the ones indicated for QAM.

For example, a SM_{11} array[4]

$$\begin{bmatrix} -a & -b & 1 \\ 1 & 0 & 0 \\ c & e & 1 \end{bmatrix}$$

---

[4] See Ref. [25] for the definition of canonical arrays.

for QAM:

$$e_1 = 2\ h/2 = h$$

$$e_2 = 0\ h/2 = 0$$

$$\varepsilon = 2\ h/2 = h$$

for QAA:

$$e_1 = 1\ h/2 = h/2$$

$$e_2 = 0\ h/2 = 0$$

$$\varepsilon = 1\ h/2 = h/2$$

Using equations (5.10) and (5.11), it can be concluded that the error magnitude bound for QAA is one-half the value in QAM in this example.

2. Quantization Before Multiplication

In order to compare QAA with QBM another approach will be used. Define the following error vectors:

$$e_u(n-1) = u^*(n-1) - [u^*(n-1)]_q$$

$$\underline{e}_x(n-1) = \underline{x}^*(n-1) - [\underline{x}^*(n-1)]_q \qquad (5.12)$$

and assuming that these errors are introduced before each multiplication process according to the value of the error control parameters α_{ij}, β_i, γ_i and δ (where i,j = 1,2), as shown in Figure 5-6, the state equations of a finite precision D.F. can be written as

$$\begin{bmatrix} x_1^*(n) \\ x_2^*(n) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1^*(n-1) \\ x_2^*(n-1) \end{bmatrix} - \begin{bmatrix} \alpha_{11}\cdot a_{11} & \alpha_{12}\cdot a_{12} \\ \alpha_{21}\cdot a_{21} & \alpha_{22}\cdot a_{22} \end{bmatrix} \begin{bmatrix} e_{x_1}(n-1) \\ e_{x_2}(n-1) \end{bmatrix}$$

$$+ \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u^*(n-1) - \begin{bmatrix} \beta_1\cdot b_1 \\ \beta_2\cdot b_2 \end{bmatrix} e_u(n-1)$$

$$v^*(n) = \begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{bmatrix} x_1^*(n-1) \\ x_2^*(n-1) \end{bmatrix} - \begin{bmatrix} \gamma_1\cdot c_1 & \gamma_2\cdot c_2 \end{bmatrix} \begin{bmatrix} e_{x_1}(n-1) \\ e_{x_2}(n-1) \end{bmatrix}$$

$$+ d\, u^*(n-1) - \delta\, d\, e_u(n-1)$$

or in vector notation

$$\underline{x}^*(n) = \underline{A}\, \underline{x}^*(n-1) - \underline{\alpha A}\, \underline{e}_x(n-1) + \underline{B}\, u^*(n-1) - \underline{\beta B}\, e_u(n-1) \tag{5.13}$$

$$v^*(n) = \underline{C}\, \underline{x}^*(n-1) - \underline{\gamma C}\, \underline{e}_x(n-1) + d\, u^*(n-1) - \delta d\, e_u(n-1)$$

If quantization after addition (QAA) is to be considered, all error control parameters ($\alpha,\beta,\gamma,\delta$) are set

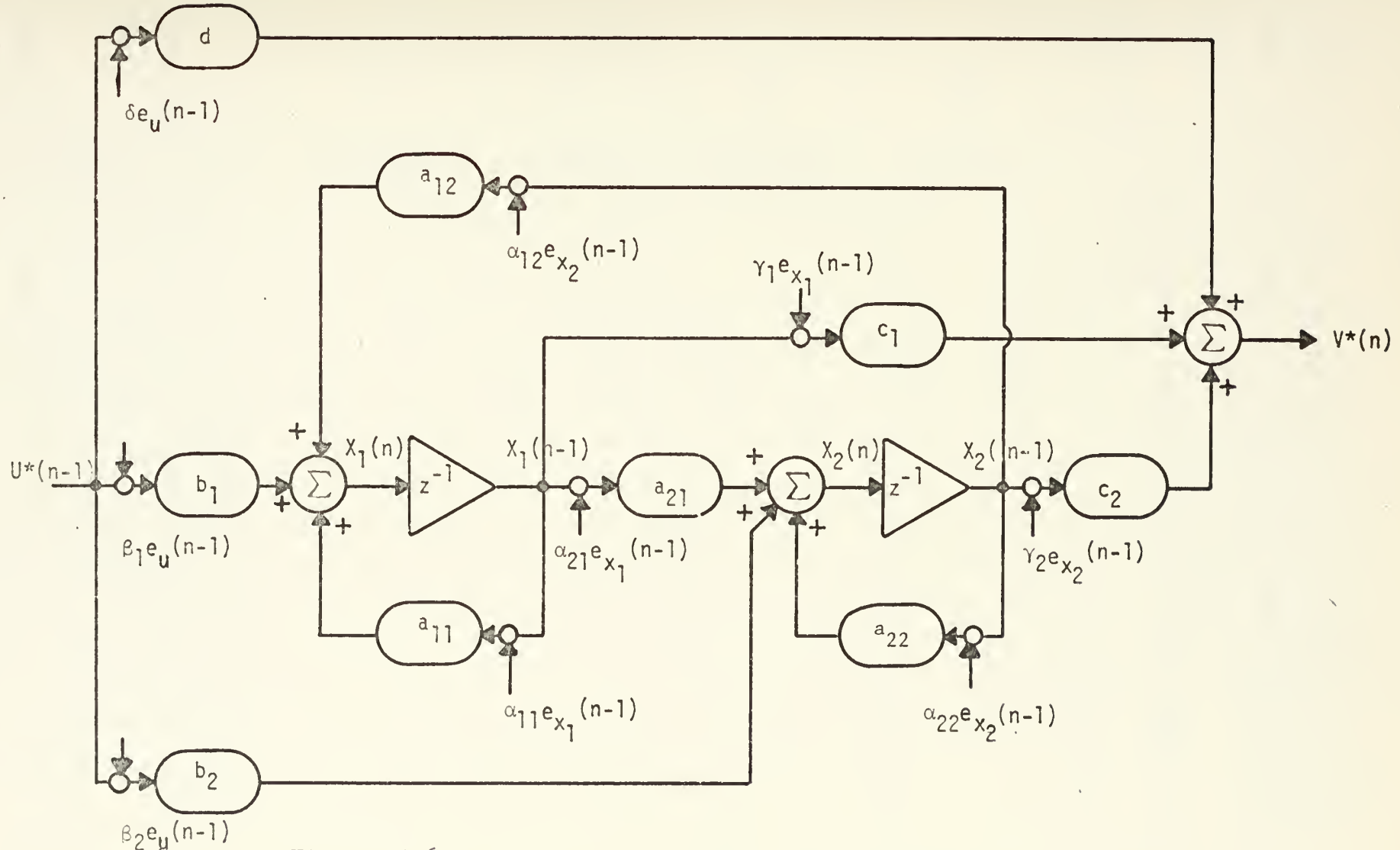


Figure 5-6. Second Order Single-Input Single-Output Digital Filter. Quantization Before Multiplication

equal to one. This is equivalent to introduce the error after the delay operator, rather than after the addition, but the value of the error is not affected.

If quantization before multiplication (QBM) is to be studied, then set

$$\alpha_{ij} = \begin{cases} 0 & \text{if } a_{ij} = 1 \\ 1 & a_{ij} \neq 1 \end{cases}$$

$$\beta_i = \begin{cases} 0 & \text{if } b_i = 1 \\ 1 & b_i \neq 1 \end{cases} \qquad (5.14)$$

$$\gamma_i = \begin{cases} 0 & \text{if } c_i = 1 \\ 1 & c_i \neq 1 \end{cases}$$

$$\delta = \begin{cases} 0 & \text{if } d = 1 \\ 1 & d \neq 1 \end{cases}$$

Here, the input signal has not been assumed to be quantized, since the output signal is not generally quantized. Therefore, these stages can also be cascade.

These error vectors are bounded, and assuming again rounding with a quantization step of $\pm h/2$, it follows that

$$e_{x_1}(n-1) = \begin{cases} h/2 & \text{if at least one of the coefficients } (a_{11}, a_{21}, c_1) \text{ is different from 0 or 1} \\ 0 & \text{otherwise} \end{cases}$$

$$e_{x_2}(n-1) = \begin{cases} h/2 & \text{if at least one of the coefficients } (a_{12}, a_{22}, c_2) \text{ is different from 0 or 1} \\ 0 & \text{otherwise} \end{cases}$$

$$e_u(n-1) = \begin{cases} h/2 & \text{if at least one of the coefficients } (b_1, b_2, d) \text{ is different from 0 or 1} \\ 0 & \text{otherwise} \end{cases}$$

Then it is possible to find constant error vectors $\underline{e}_x$ and e_u whose elements are larger than the magnitude of the corresponding elements of $\underline{e}_x(n-1)$ and $e_u(n-1)$, or

$$|\underline{e}_x(n-1)| \leq \underline{e}_x$$

and

$$|e_u(n-1)| \leq e_u.$$

It can be observed that the value of this constant vector component depends on the existence of nonzero non-ones columns on the D.F. array, rather than on the rows.

$$e_{x_k} = \nu_k \frac{h}{2} \qquad k = 1,2 \tag{5.15}$$

$$e_u = \nu_3 \frac{h}{2}$$

where

$$\nu_k = \begin{cases} 0 & \text{if all elements in the Kth column of the D.F. array are 0 or 1} \\ 1 & \text{otherwise} \end{cases} \qquad k=1,2,3$$

Defining the state and the output errors as before, it results in

$$\underline{y}(n) = \underline{x}(n) - \underline{x}^*(n)$$

$$= \underline{A}\,\underline{x}(n-1) + \underline{B}\,u^*(n-1)$$

$$- [\underline{A}\,\underline{x}^*(n-1) - \underline{\alpha A}\,\underline{e}_x(n-1) + \underline{B}\,u^*(n-1) - \underline{\beta B}\,e_u(n-1)]$$

$$= \underline{A}\,\underline{y}(n-1) + \underline{\alpha A}\,\underline{e}_x(n-1) + \underline{\beta B}\,e_u(n-1) \tag{5.16}$$

$$\Delta v(n) = v(n) - v^*(n)$$

$$= \underline{C}\,\underline{x}(n-1) + d\,u^*(n-1)$$

$$- [\underline{C}\,\underline{x}(n-1) - \underline{\gamma C}\,\underline{e}_x(n-1) + d\,u^*(n-1) - \delta d\,e_u(n-1)]$$

$$= \underline{C}\,\underline{y}(n-1) + \underline{\gamma C}\,\underline{e}_x(n-1) + \delta d\,e_u(n-1) \tag{5.17}$$

Assuming $\underline{x}(-1) = \underline{x}^*(-1)$ and $\underline{e}_x(-1) = e_u(-1) = 0$, and using the propagation error equation (5.16),

$$\underline{y}(0) = \underline{x}(-1) - \underline{x}^*(-1) = 0$$

$$\underline{y}(1) = \underline{A}\,\underline{y}(0) + \underline{\alpha A}\,\underline{e}_x(0) + \underline{\beta B}\,e_u(0) = \underline{\alpha A}\,\underline{e}_x(0) + \underline{\beta B}\,e_u(0)$$

$$\underline{y}(2) = \underline{A}\,\underline{y}(1) + \underline{\alpha A}\,\underline{e}_x(1) + \underline{\beta B}\,e_u(1)$$

$$= \underline{A}\,\underline{\alpha A}\,\underline{e}_x(0) + \underline{\alpha A}\,\underline{e}_x(1) + \underline{A}\,\underline{\beta B}\,e_u(0) + \underline{\beta B}\,e_u(1)$$

then

$$\underline{y}(n) = \sum_{\ell=0}^{n-1} \underline{A}^{n-\ell-1}\left[\underline{\alpha A}\,\underline{e}_x(\ell) + \underline{\beta B}\,e_u(\ell)\right] \tag{5.18}$$

$$= \sum_{\ell=0}^{n} \underline{A}^{\ell}\left[\underline{\alpha A}\,\underline{e}_x(n-\ell-1) + \underline{\beta B}\,e_u(n-\ell-1)\right] \tag{5.19}$$

and from equation (5.17) using (5.18) and (5.19)

$$\Delta v(n) = \underline{C} \sum_{\ell=0}^{n-2} \underline{A}^{n-\ell-2}\left[\underline{\alpha A}\,\underline{e}_x(\ell) + \underline{\beta B}\,e_u(\ell)\right] + \underline{\gamma C}\,\underline{e}_x(n-1) + \delta d\,e_u(n-1) \tag{5.20}$$

$$= \underline{C} \sum_{\ell=0}^{n-1} \underline{A}^{\ell}\left[\underline{\alpha A}\,\underline{e}_x(n-\ell-2) + \underline{\beta B}\,e_u(n-\ell-2)\right] + \underline{\gamma C}\,\underline{e}_x(n-1)$$

$$+ \delta d\,e_u(n-1) \tag{5.21}$$

From equation (5.19) it follows that the state error magnitude vector is

$$\langle \underline{y}(n) \rangle = \left\langle \sum_{\ell=0}^{n} \underline{A}^{\ell}\, \underline{\alpha A}\, \underline{e}_x(n-\ell-1) + \underline{A}^{\ell}\, \underline{\beta B}\, e_u(n-\ell-1) \right\rangle$$

$$\leq \sum_{\ell=0}^{n} \langle \underline{A}^{\ell} \rangle \langle \underline{\alpha A} \rangle\, \underline{e}_x + \langle \underline{A}^{\ell} \rangle \langle \underline{\beta B} \rangle\, e_u \qquad (5.22)$$

and from equation (5.21) using (5.17) the output error magnitude bound can be obtained

$$\langle \Delta v(n) \rangle \leq \langle \underline{C} \rangle \langle \underline{y}(n-1) \rangle + \langle \underline{\gamma C} \rangle\, \underline{e}_x + |\delta d|\, e_u \qquad (5.23)$$

where the state error bound, $\langle \underline{y}(n-1) \rangle$, is given by equation (5.22).

As observed previously, for QAA all error control parameters are equal to unity. Therefore for QAA, equation (5.22) and (5.23) reduces to

$$\langle \underline{y}(n) \rangle \leq \langle \underline{A}^{\ell+1} \rangle\, \underline{e}_x + \langle \underline{A}^{\ell} \rangle \langle \underline{B} \rangle\, e_u \qquad (5.24)$$

$$\langle \Delta v(n) \rangle \leq \langle \underline{C} \rangle \langle \underline{y}(n-1) \rangle + \langle \underline{C} \rangle\, \underline{e}_x + |d|\, e_u \qquad (5.25)$$

For QBM, it holds that

$$\langle \underline{\alpha A} \rangle \leq \langle \underline{A} \rangle$$

$$<\underline{\beta B}> \leq <\underline{B}>$$

$$<\underline{\gamma C}> \leq <\underline{C}>.$$

$$|\delta d| \leq |d|$$

since $<\underline{Q}>$ is defined as the matrix formed by the absolute value of each element of the matrix $\underline{Q}$.

Therefore the bounds for QBM given by equations (5.22) and (5.23) are at least as large as the bounds given for QAA by equations (5.24) and (5.25), respectively.

E. CONCLUSIONS

Quantization after addition and quantization before multiplication methods have been shown applicable to hardware implementation of digital filters. Advantages of these two methods over the usual quantization after multiplication has been demonstrated and QBM proved to be the more effective to reduce error quantization bounds. Therefore QBM is the most suitable form for hardware implementation of digital filters. The modification required to perform rounding or truncation before multiplication using the available NRMEC chips has been presented.

APPENDIX A

POLE-ZERO CORRESPONDENCE IN S AND Z-DOMAIN

1. Definition of the Z-Transform

Given a sequence $\{x(n)\}_{n=\infty}^{\infty}$ the two-sided z-transform is defined as

$$X(z) \overset{\Delta}{=} Z[x(n)] = \sum_{n=-\infty}^{\infty} x(n)\, z^{-n} \quad . \tag{A.1}$$

When $x(n) = 0$ for $n < 0$, the one-sided z-transform is defined

$$X(z) \overset{\Delta}{=} \sum_{n=0}^{\infty} x(n)\, z^{-n} \tag{A.2}$$

From the relation to the Laplace-Fourier transform

$$z^{-1} = e^{-sT} \tag{A.3}$$

is called the unit delay operator.

2. Mapping S-Plane into Z-Plane

Breaking s and z into real and imaginary parts,

$$s = \sigma + j\omega \quad \text{and} \quad z = \alpha + j\nu$$

Since

$$z = e^{Ts} = e^{T\sigma} \quad e^{j\omega T} = \underbrace{e^{T\sigma\cos\ \omega T}}_{\alpha} + j\underbrace{e^{T\sigma\sin\ \omega T}}_{\nu} \qquad (A.4)$$

When, $\omega = 0$, then from (A.4)

$$\nu = 0 \quad \text{and} \quad \alpha = e^{T\sigma} \quad \begin{matrix} > 1 & \text{for} & \sigma > 1 \\ < 1 & \text{for} & \sigma < 1 \end{matrix}$$

For a pole at $-\infty$, $\sigma = -\infty$, and from (A.4) $\nu = 0$ and $\alpha = 0$, then mapp onto the origin of the z plane.

For imaginary poles, $\sigma = 0$, we have from (A.4) $\nu = \sin \omega t$ and $\alpha = \cos \omega t$ or $\nu^2 + \alpha^2 = 1$, therefore the imaginary axis of the s plane maps on the unit circle of the z plane.

Figure A-1 summarizes the mapping of the s plane into the z plane. The left half s plane is mapped inside the unit circle ($|z| = 1$) in the z plane. The imaginary s plane is mapped onto $|z| = 1$. The right half s plane is mapped into the region $|z| > 1$. The left stripe/limited by half the sampling frequency ($\pm\omega_s/4$) in the s plane maps to the right within $|z| < 1$ region. The left stripes bounded by $+\omega_s/4$ and $+\omega_s/2$ or $-\omega_s/4$ and $-\omega_s/2$ in the s plane maps to the left within $|z| < 1$ region. The point at infinity in the negative real s-plane is mapped into the z-plane origin, and the s-plane origin is mapped into the +1 point in the z-plane. It can be concluded that the farther the real component of the s-plane complex pole is

located from the imaginary axis, the closer the z-plane complex pole is to the origin, which means the faster the discrete output sequence will converge, i.e., the damping is more pronounced.

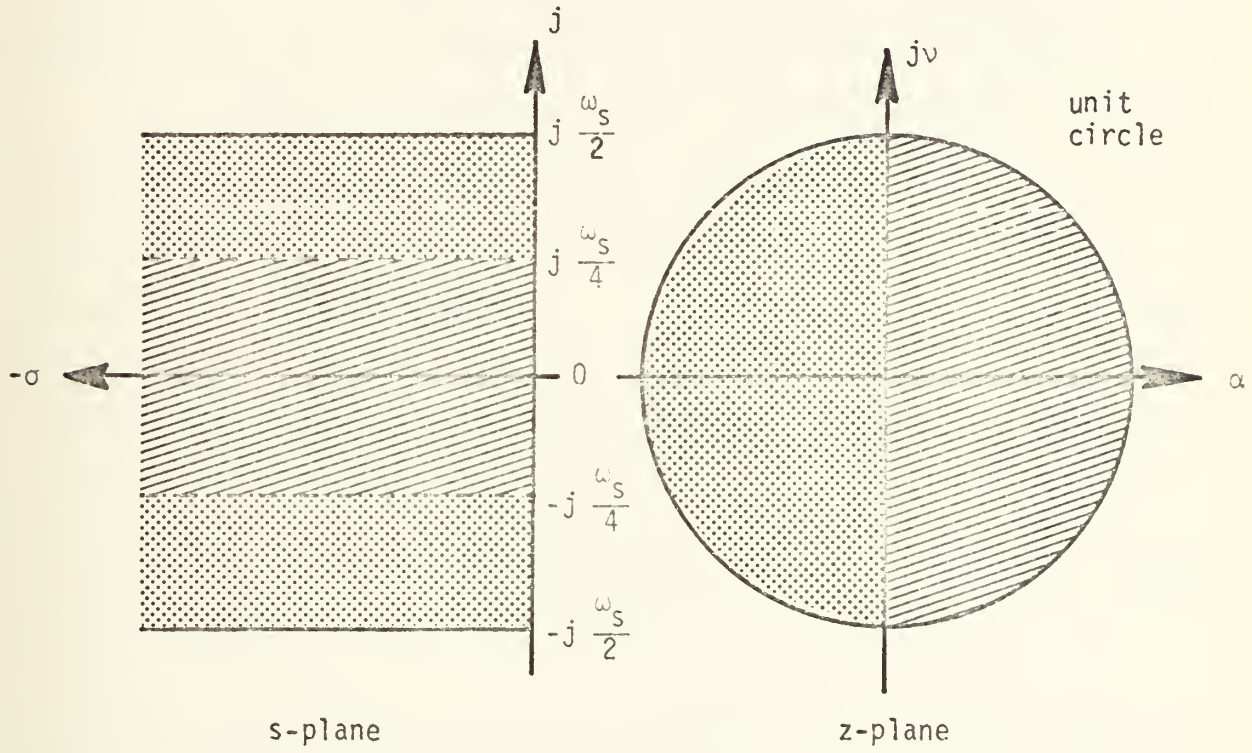


Figure A-1. Mapping s-Plane into z-Plane

APPENDIX B

DISCRETE TRANSFER FUNCTION REALIZATION

1. Discrete Transfer Functions

A linear time-invariant discrete-time filter is described by the difference equation

$$y(nT) = \sum_{k=0}^{M} a_k x[(n-k)T] - \sum_{k=1}^{N} b_k y[(n-k)T] \qquad (B.1)$$

which discrete output, y(nT), is a linear combination of the past and present M input samples and N output samples.

The transfer function of this discrete system, similarly as for the continuation case is defined

$$G(z) \overset{\Delta}{=} \frac{Y(z)}{X(z)} \qquad (B.2)$$

and taking the z-transform of (B.1) and rearranging gives

$$G(z) = \frac{\sum_{k=0}^{M} a_k z^{-k}}{1 + \sum_{k=1}^{N} b_k z^{-k}} \qquad (B.3)$$

The observation of this transfer function shows that it is identical to those obtained from the Laplace transform analysis of continuous systems described by linear constant coefficients, ordinary differential equations. The roots of the denominator of G(z) are called the poles of the discrete

system, and the roots of the denominators are called the zeros. However, the discrete system is stable in the sense that every bounded input sequence yields a bounded output sequence if and only if the poles of G(z) lie within the unit circle in the z-plane.

The frequency spectrum of the discrete system is periodic in ω with period $2\pi/T$ due to sampling, and this spectrum can be computed by letting $z = \exp(j\omega T)$ in the transfer function.

2. Recursive Filter Realization

If in the transfer function of a D.F., all b_k are zero, the filter has no feedback, as revealed by inspection of (B.1) or (B.3), and is said to be of the nonrecursive or transversal type.

If at least one b_k and one a_k value are nonzero, the filter is called recursive.

The nonrecursive filter has finite memory and can have excellent phase characteristics, but tends to require a large number of terms to obtain a relative sharp cut off [16]. The recursive filter has an infinite memory and tends to have fewer terms. Therefore sharp cut off filters are much easier to design using a recursive structure. The design method for this type of filter will be discussed later.

A transfer function can be realized by direct form or by reduction to lower order form, generally first or second order sections in a cascade, parallel or hybrid structure.

a. Direct Realization

From a given transfer function of a D.F. the difference equation (B.1) can be obtained, and performing the direct operations implied by that equation the so called "direct" realization is obtained, as shown in Figure B-1.

Using an intermediate variable w(n) such that

$$w(nT) = \sum_{k=1}^{N} b_k w[(n-k)T] + x(nT)$$

equation (B.1) can be written

$$y(nT) = \sum_{k=0}^{M} a_k w[(n-k)T] \qquad (B.4)$$

The realization based upon (B.4) is shown in Figure B-2 and is called the "canonical" realization of the filter, since the number of delays and multipliers is minimized.

b. Reduction to Lower Order Forms

This form is more convenient because lower order forms present not only a smaller coefficient sensitivity [16] but also a reduced quantization noise effect [18]. Thus, a higher order filter is obtained by combining first and second order sections.

(1) Cascade Realization

By factoring the overall transfer function can be written associating zeros and poles in the form

$$H(z) = k_p + \sum_{i=1}^{p} G_i(z) \qquad (B.5)$$

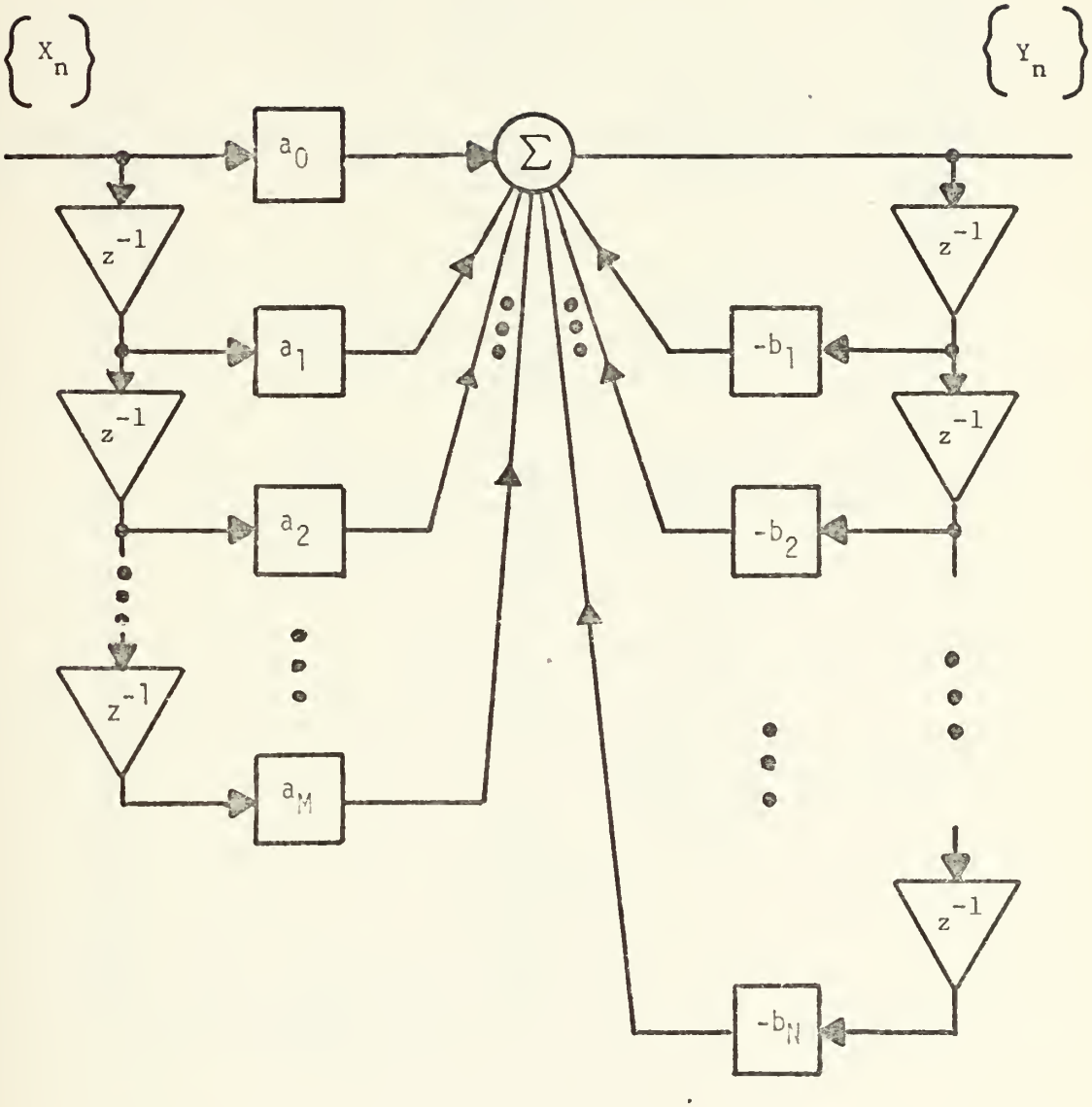


Figure B-1. Direct Realization

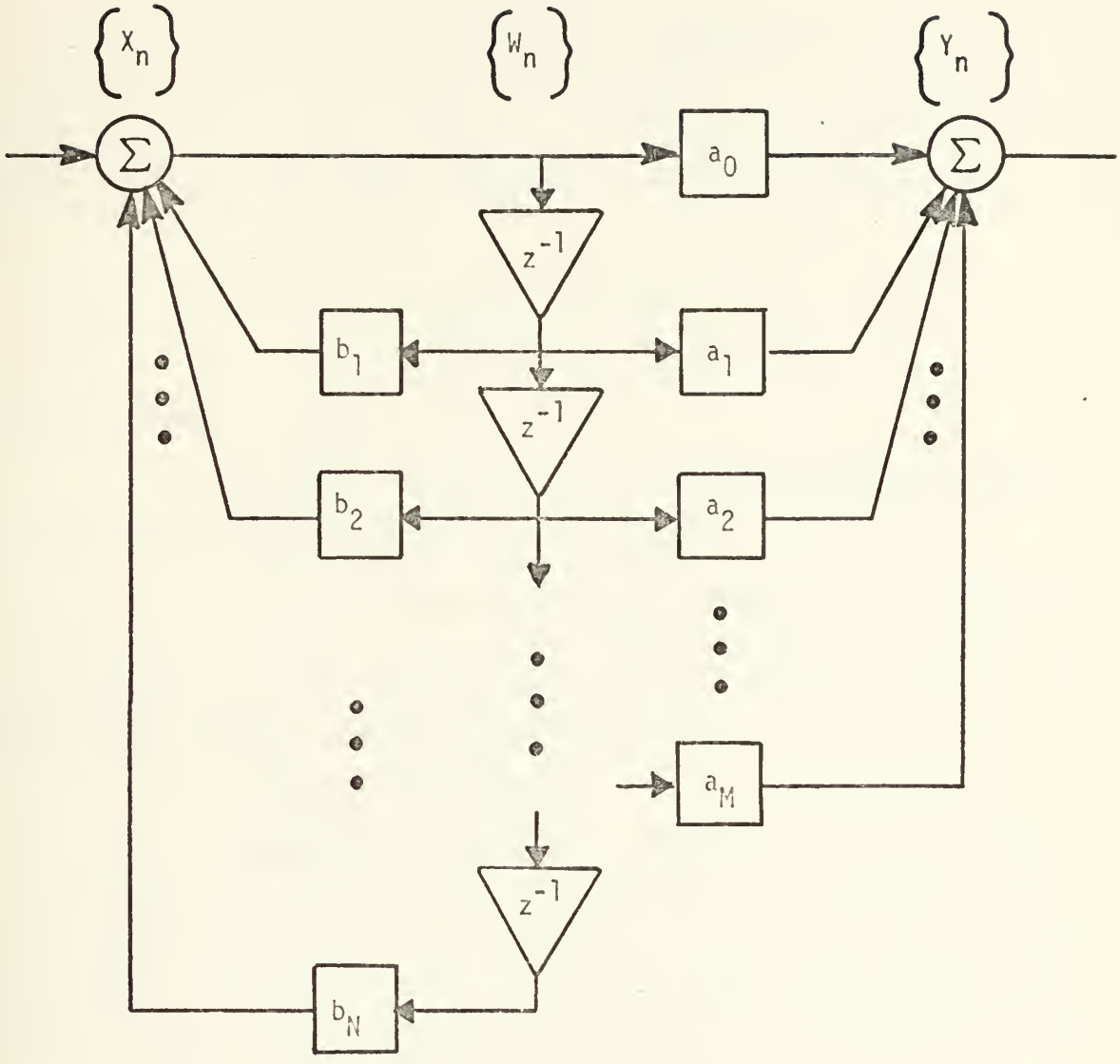


Figure B-2. Canonical Realization

as illustrated in Figure B-3, where $G_i(z)$ represents the transfer function of the first or second order sections.

(2) Parallel Realization

By partial fraction expansion, a transfer function with simple poles can be written as the sum of the first and second order transfer functions, in the form

$$H(z) = k_c \prod_{i=1}^{c} G_i(z)$$

as realized in Figure B-4.

If the transfer function has multiple poles, higher order sections will be required.

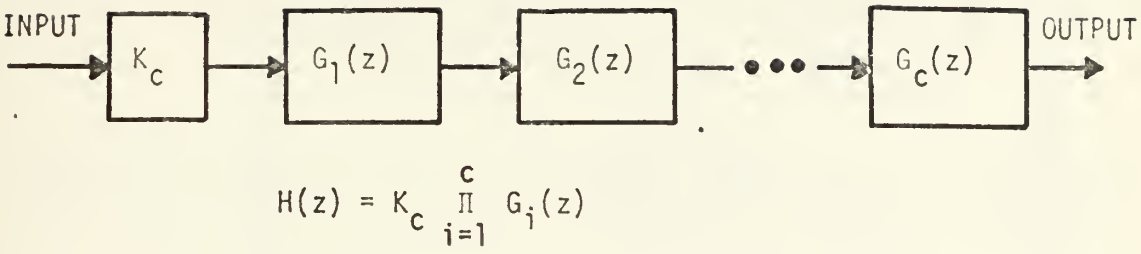
The parallel realization permits an easy scaling of the D.F., but the obtaining of the transfer function and the zeros are not readily identifiable.

(3) Hybrid Realization

The hybrid form is a combination of parallel and cascade, as shown in Figure B-5 (a) and (b) the design to obtain the hybrid form is not as simple and should only be used when the cascade form becomes too difficult to scale.

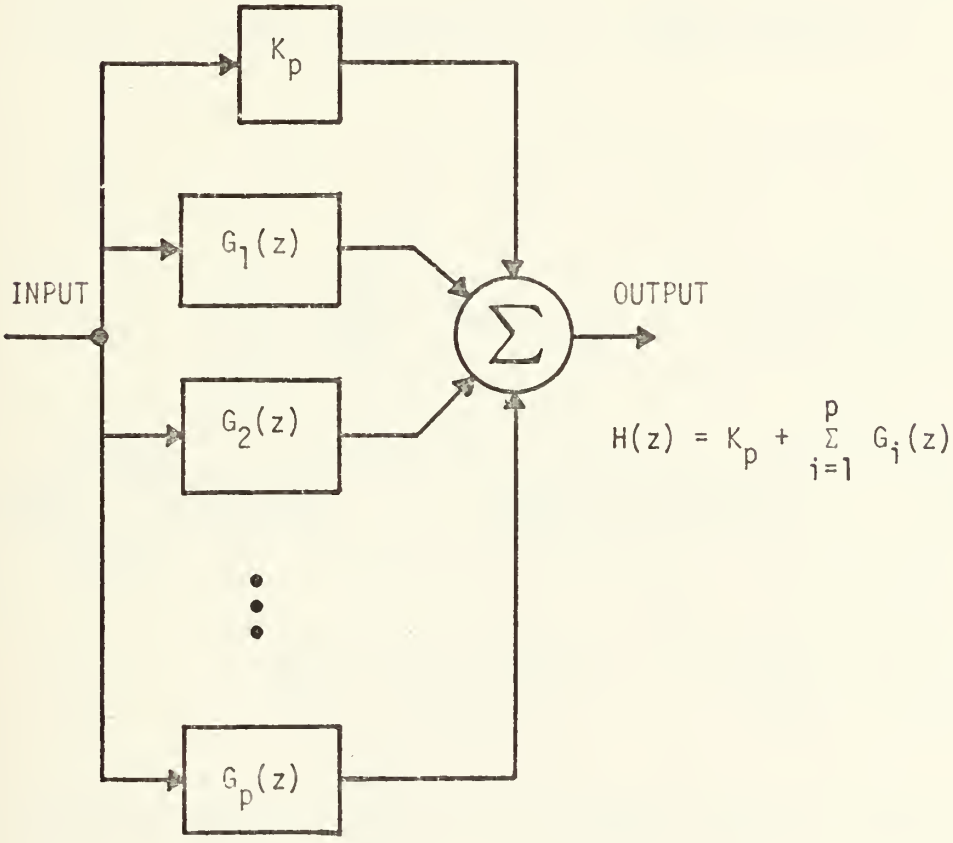
3. Nonrecursive Filter Realization

The z-transform applied to a continuous filter transfer function can not be applied to nonrecursive filters, also called transversal filters. This type of filter is very useful, in particular, if a linear phase minimum phase or a prescribed magnitude characteristic is desired.



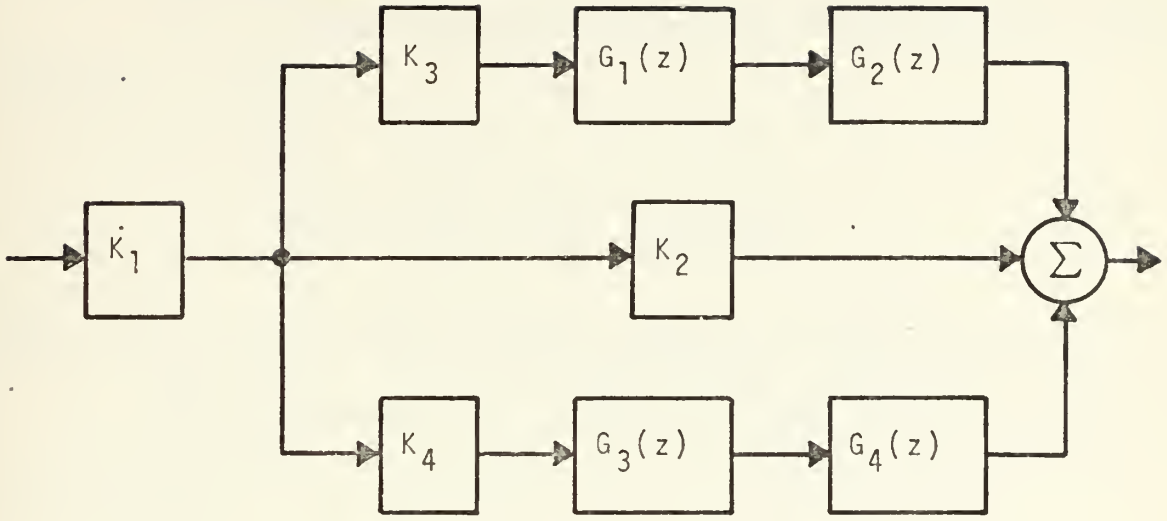
$$H(z) = K_c \prod_{i=1}^{c} G_i(z)$$

Figure B-3. Cascade Realization of H(z)



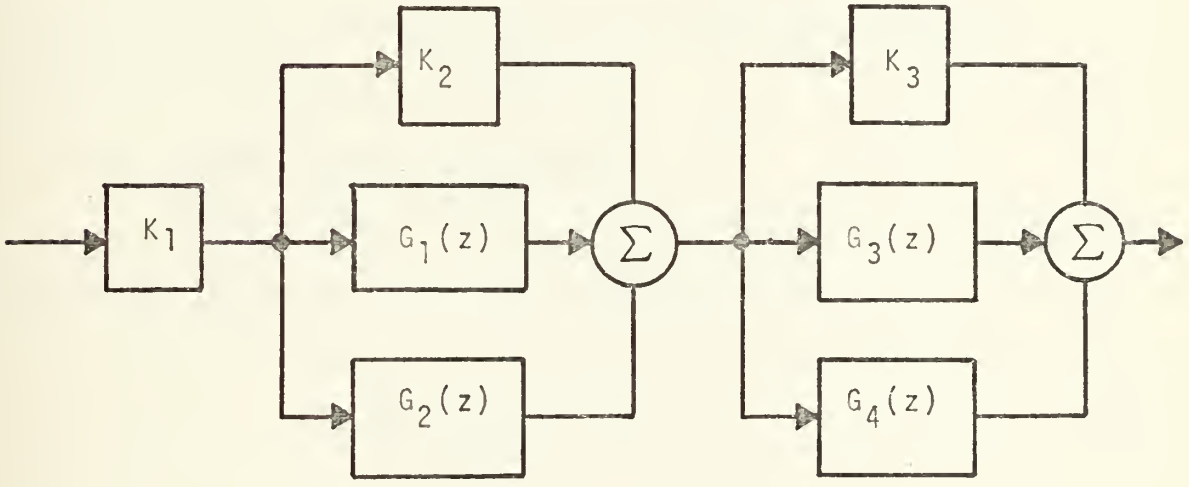
$$H(z) = K_p + \sum_{i=1}^{p} G_i(z)$$

Figure B-4. Parallel Realization of H(z)



$$H(z) = K_1[K_2 + K_3G_1(z)G_2(z) + K_4G_3(z)G_4(z)]$$

(a)



$$H(z) = K_1[K_2 + G_1(z) + G_2(z)][K_3 + G_3(z) + G_4(z)]$$

(b)

Figure B-5. Hybrid Realizations

a. Convolution Approach

For a linear discrete system the following convolution summation applies

$$y(nT) = \sum_{m=0}^{M} x(mT)\, h[(n-m)T] \qquad (B.7)$$

where h[(n-m)T] is the discrete impulse response delayed mT.

From equation (B.7) a discrete time transfer function is obtained

$$G(z) = \frac{Y(z)}{X(z)} = \sum_{\ell=0}^{\infty} h(\ell T)\, z^{-\ell} \qquad (B.8)$$

$$= h(0) + h(T)\, z^{-1} + h(2T)\, z^{-2} + \ldots \qquad (B.9)$$

which leads to a nonrecursive or transversal filter realization shown in Figure B-6.

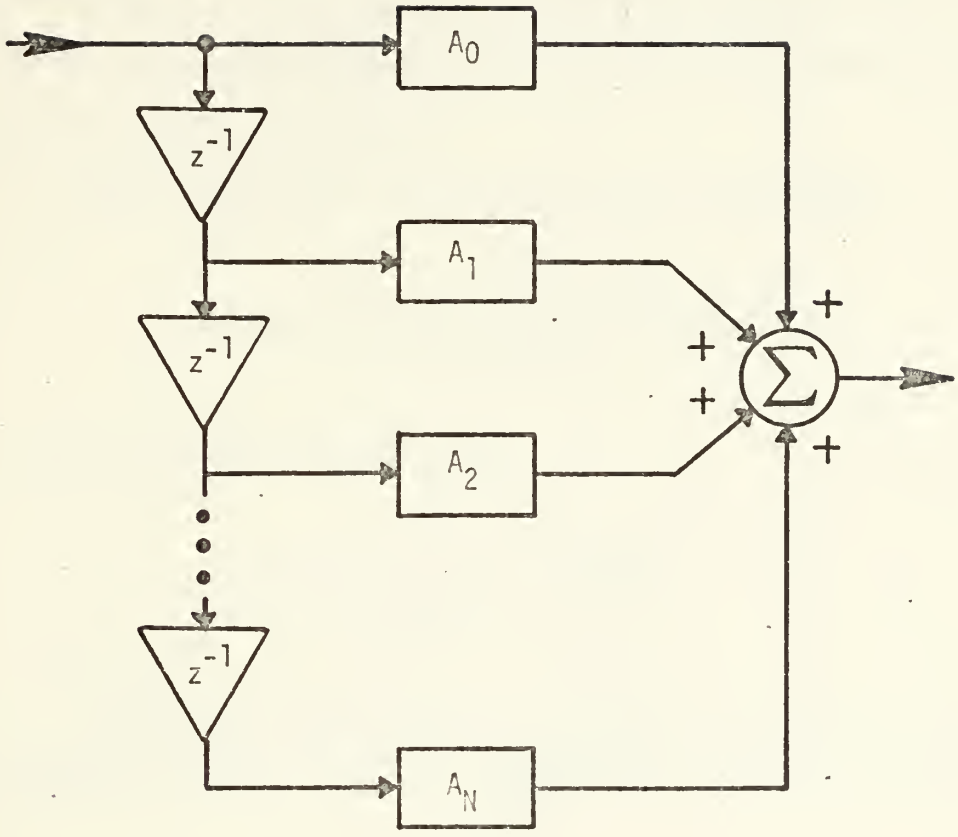
b. Fourier Series Approach

As mentioned before, a nonrecursive filter has all b_k equal to zero. Then from equation (B.3)

$$G(z) = \sum_{k=0}^{\infty} a_k\, z^{-k} \qquad (B.10)$$

letting M generically go to infinity. Equations (B.8) and (B.10) are equivalents.

Due to sampling the frequency response of a discrete time filter is periodic, with period equal to



$$H(z) = \sum_{n=0}^{N} A_n z^{-n}$$

Figure B-6. Block Diagram of a Non-Recursive or Transversal Filter

its sampling frequency, $\omega_s = 2\pi/T$. This periodic frequency response may be represented as a Fourier series. The form of the series to be used will depend on whether the desired frequency characteristics are an odd or even function with respect to zero frequency.

Even functions can be written in the form

$$G_e(j\omega) = A_0 + \sum_{n=1}^{\infty} A_n \cos(wnT) \tag{B.11}$$

and odd functions in the form

$$G_o(j\omega) = \sum_{n=1}^{\infty} B_n \sin(wnT) \tag{B.12}$$

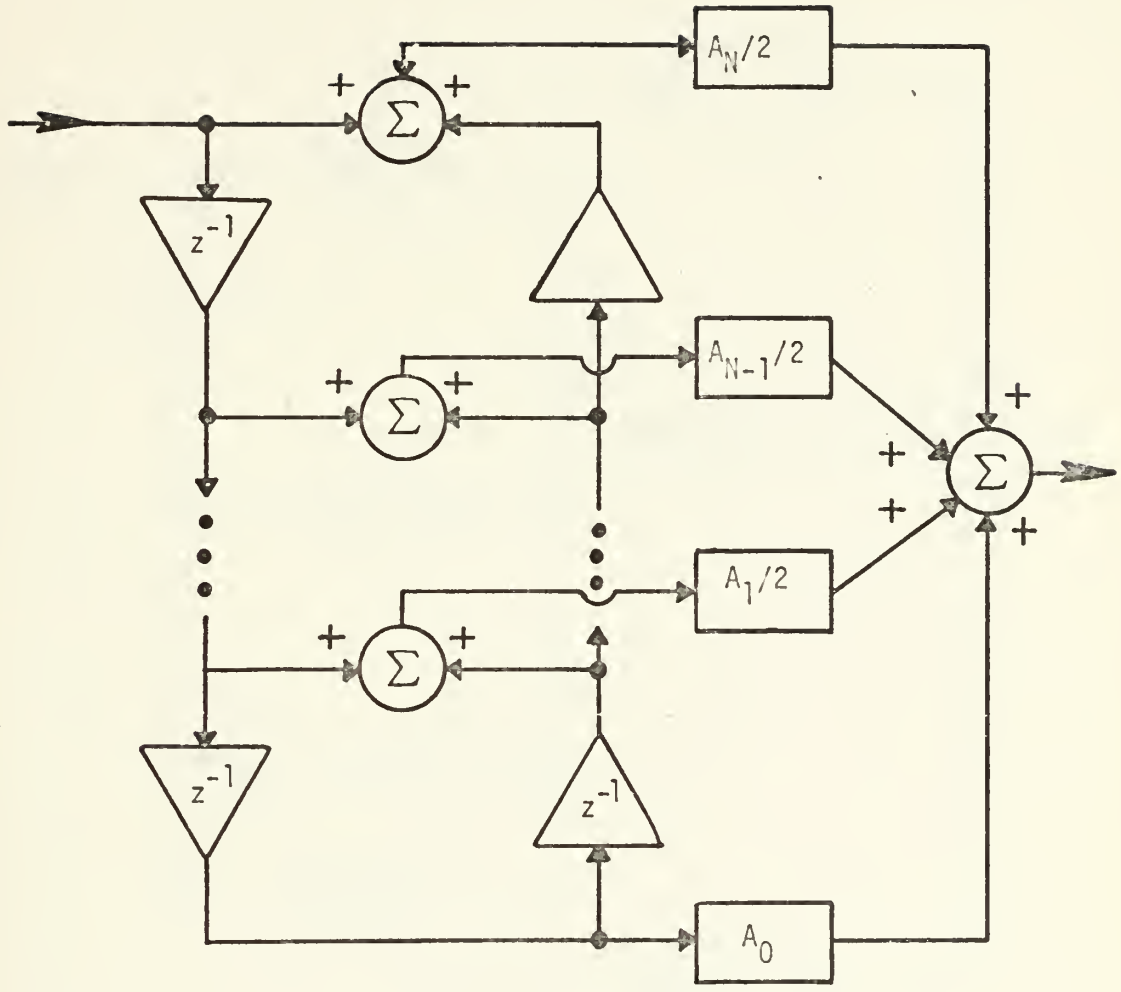
Using the relation $z = \exp(j\omega T)$, equations (B.11) and equation (B.12) can be presented as (B.13) and (B.14).

$$G_e(j\omega) = A_0 + \sum_{n=1}^{\infty} \frac{A_n}{2} (z^n + z^{-n}) \tag{B.13}$$

$$G_o(j\omega) = \sum_{n=1}^{\infty} \frac{B_n}{2j} (z^n - z^{-n}) \tag{B.14}$$

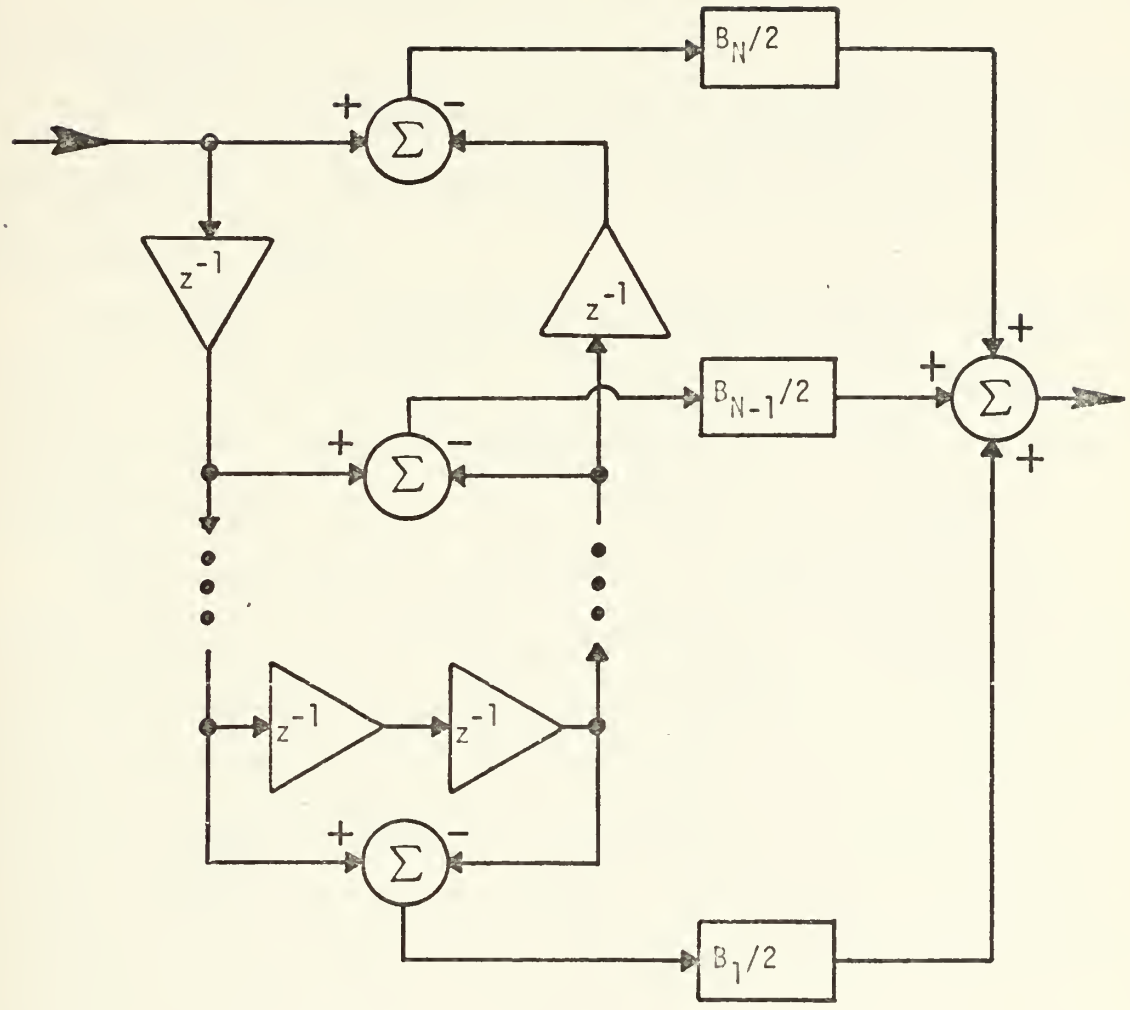
To obtain filters with real coefficients, the j of equation (B.14) can be dropped. The resulting filter will have a phase shift displacement of 90° from the theoretical function, but the magnitude function will not be affected.

Figures B-7 and B-8 illustrate the block diagram realization of nonrecursive filters for finite (since the



$$H(z) = z^{-N}\left[A_0 + \sum_{n=1}^{N} \frac{A_N}{2}\left(z^n + z^{-n}\right)\right]$$

Figure B-7. Block Diagram of Transversal Filter Mechanization for Finite Fourier Cosine Series



$$H(z) = z^{-N} \left[\sum_{n=1}^{N} \frac{B_n}{2} (z^n - z^{-n}) \right]$$

Figure B-8. Block Diagram of Transversal Filter Mechanization for Finite Fourier Sine Series

summation stops after N terms) Fourier cosine and sine series, respectively.

3. Windowing

In order to establish a physical realizable filter design, the summation in equations (B.8), (B.13) and (B.14) must stop after N terms.

The effect of truncating the response from an infinite number of terms accounts to a distortion of the frequency response curve, called "GIBB's phenomenon", which is what normally happens when a Fourier series is truncated.

This truncation is equivalent to multiplication by a window function, W_k, which is nonzero for a length of time NT, or in the frequency domain is equivalent to the convolution $G'(w) = G(w)*W(w)$. This accounts for the distortion in the frequency domain, but also helps to avoid it, if a proper window function is chosen. In general, a low pass filtering or smoothing of the magnitude response is obtained by the window function.

The best known are the Haming and Hanning windows [14]. The Kaiser window [16] is relatively easy to use and exhibits superior side lobe suppression and produces designs which compare with others developed through more involved procedures [3].

APPENDIX C

FUNCTIONAL TRANSFORMS

There are three most common methods of mapping a transfer function from the s-domain to the z-domain: standard, bilinear and matched z transforms.

The bilinear and matched are optimized for sine waves yielding the most accurate transform in the communications field. A summary of each transform is presented next and a comparison table is shown in Figure C-1.

The hand calculation of these transforms for more than a first-order one stage filter is extremely complex and requires a high level of accuracy. Therefore the use of a computer program [13] is helpful.

1. Standard z-Transform

The standard or impulse invariant z-transform uses the transformation $z = \exp(sT)$. It requires the partial fraction expansion of the transfer function of the continuous filter. Therefore a sum of first order terms is obtained and the exponential transform indicated on Figure C-1 is applied to each one, yielding a parallel realization. In general, this representation gives excellent results when applied to all-pole low-pass and bandpass filters [12]. The design of bandstop and high-pass filters can only be accomplished adding in cascade a wideband low-pass filter, called "guard filter" in order to eliminate folding.

2. Bilinear z-Transform

The bilinear z-transform (trapezoidal integration) eliminates the folding problem of standard z-transforms, and is very useful to realize digital filters that have relative constant magnitude passband and stopband characteristics.

This transformation

$$s = (2/T)\ (1-z^{-1})/(1 + z^{-1})$$

is an algebraic one, so it can be applied to the factored or unfactored transfer function of the continuous filter. This mapping, however, distorts the frequency response. Therefore it is necessary to counter-wrap the desired radian frequency response before applying the transformation. Then each critical imaginary frequency ω_i is replaced by $2/T\ \mathrm{Tan}(1/2\omega_i T)$. This still does not yield an exact equivalence between the two frequency responses, therefore care must be used when designing filters with critical frequencies near the half-sampling frequency.

3. Matched z-Transform

This transformation generates a digital transfer function with poles and zeros matched to those of the continuous functions. The exponential transformation $s = \exp(sT)$ is then applied to poles and zeros. It requires factoring both

numerator and denominator of the continuous transfer function to the form s - b and replaced by $1 - z^{-1} \exp(bT)$. Additional zeros at half the sampling frequency may be required in order that the power of the poles and zeros are the same.

Standard Z-Transform	Bilinear Z-Transform	Matched Z-Transform
Yields a Parallel Realization	Yields Parallel or Series Realization	Yields a Series Realization
Requires Partial Fraction Expansion of Filter Transfer Function	Requires Pre-Warped Filter Transfer Function	Requires Factored Form of Filter Transfer Function
Preserves Shape of Impulse-Time Response	Preserves Flat Magnitude Gain-Frequency Response Characteristics	Preserves Shape of Frequency Response Characteristics
Suitable for Bandlimited Functions (lowpass and Bandpass) Only	Suitable for all Filter Types Especially Wide Bandwidth Filters	Suitable for all Types but may require insertion of Additional Zeros at the Half-Sampling Frequency
Exponential Transform	Algebraic Transform	Exponential Transform
$\frac{R}{s-b} \longrightarrow \frac{RT}{1-z^{-1}\exp(bT)}$	$s-b \longrightarrow \frac{2}{T}\frac{(1-z^{-1})}{(1+z^{-1})} - b_1$	$s-b \longrightarrow 1 - z^{-1}\exp(bT)$

Table C-1. Comparison of the three types of z-transforms available to transform poles and zeros (transfer function) from the s-plane to the z-plane

APPENDIX D

AMPLITUDE BOUND OF LIMIT CYCLES IN D.F. USING LYAPUNOV's DIRECT METHOD

For the case of quantization after addition (QAA) a second-order digital filter section with two poles and no zeros will be studied similarly and compared with results obtained by Parker and Hess [1] for quantization after multiplication (QAM).

The system presented in Figure D-1 for QAM can be redrawn as shown in Figure D-2 considering roundoff after addition and described by the following difference equation (where u(n) = 0)

$$x^*(n) = [-a\ x^*(n-1) - b\ x^*(n-2)]_q \qquad (D.1)$$

For a normalized quantization step (h=1), this equation can be written as

$$x^*(n) = -a\ x^*(n-1) - b\ x^*(n-2) \pm [.5 - \delta(n)] \qquad (D.2)$$

where

$$0 \leq \delta(n) \leq 1.0$$

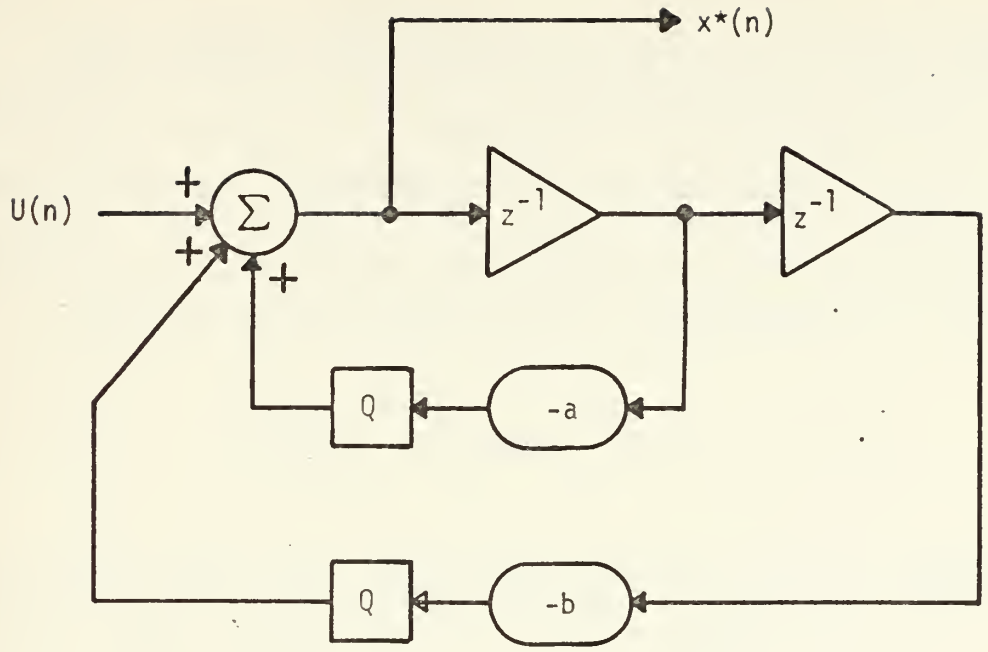


Figure D-1. Second Order D.F. with Two Poles Using Quantization After Multiplication

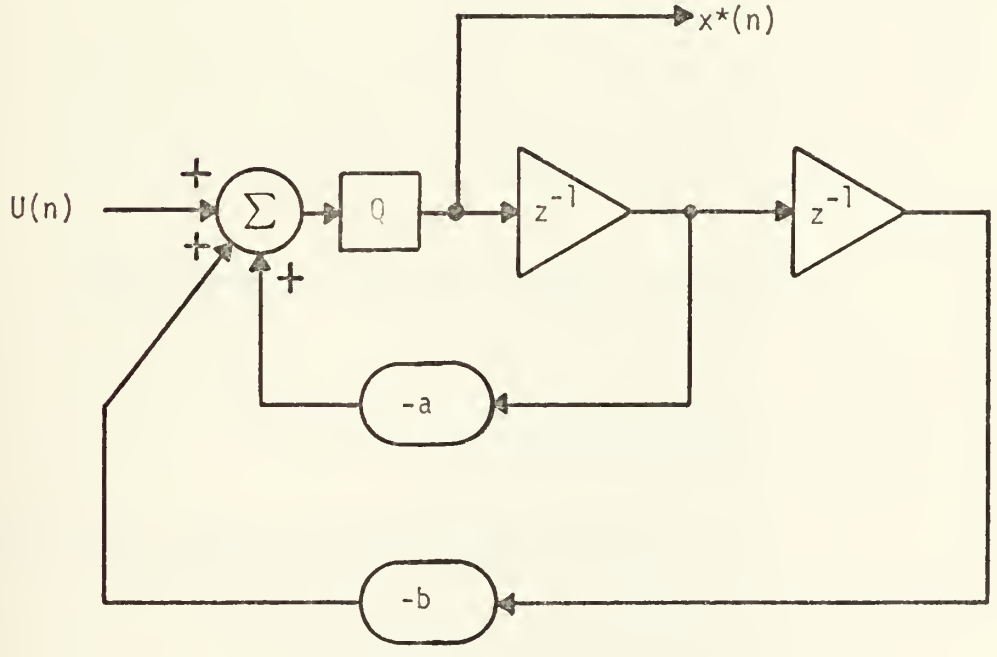


Figure D-2. Second Order D.F. with Two Poles Using Quantization After Addition

The roundoff noise sequence $e(n) = .5 - \delta(n)$ range between $\pm.5$ and can be considered as driving function to the difference equation (D.2) for the study of the natural response of the system (zero input, initial condition only).

Then the error source can be considered as an input

$$|u(n)| = |e(n)| \leq .5$$

and using the state variables $x_1^*(n) = x^*(n-2)$, $x_2^*(n) = x^*(n-1)$, $u(n) = e(n)$, equation (D.2) can be written as

$$\underline{x}^*(n+1) = \begin{bmatrix} 0 & 1 \\ -b & -a \end{bmatrix} \underline{x}^*(n) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(n) \qquad (D.3)$$

or

$$\underline{x}^*(n+1) = \underline{A}\, \underline{x}^*(n) + \underline{B}\, u(n) \qquad (D.4)$$

The transfer function of this filter is

$$G(z) = \frac{1}{1 + a\, z^{-1} + b\, z^{-2}} \qquad (D.5)$$

and its characteristic equation is

$$1 + a\, z^{-1} + b\, z^{-2} = 0 \qquad (D.6)$$

The steady state frequency response is obtained by setting $z^{-1} = e^{-j\omega T}$ where T is the sampling interval. Therefore,

$$0 = 1 + a(\cos \omega T - j \sin \omega T) + b(\cos 2\omega T - j \sin 2\omega T)$$

$$0 = 1 + a \cos \omega T + b \cos 2\omega T - j \sin (\omega T)(a + 2b \cos \omega T) \qquad (D.7)$$

This equation is satisfied if both real and imaginary parts are simultaneously satisfied, then the imaginary part is zero when

i) $a + 2b \cos \omega T = 0 \qquad \cos \omega T = -\frac{a}{2b}$

since $\cos 2\omega T = 2(\cos \omega T)^2 - 1 = \frac{a^2}{2b^2} - 1$
the real part becomes

$$0 = 1 + a(-\frac{a}{2b}) + b(\frac{a}{2b^2} - 1) = 1 - b \qquad (D.8)$$

ii) $\sin \omega T = 0 \qquad T = K\pi \qquad K = 0,1,2 \ldots$

the real part becomes

$$0 = 1 + (-1)^k a + b \qquad (D.9)$$

Equation (D.8) and (D.9) are the stability boundaries for a so-called "linear" second order D.F.

The term linear means that overflow or saturation arithmetic which may occur for large signal amplitude is not considered, but only the nonlinearity characteristics of the quantizer. Therefore small signal amplitudes are assumed.

Then a linear filter, as defined previously, has the stability boundaries

$$\begin{array}{ll} 1 - b = 0 & b = 1 \\ 1 \pm a + b = 0 & |a| = 1 + b \end{array} \qquad \text{(D.10)}$$

Therefore, for $b < 1$ and $|a| < (1+b)$ the corresponding linear system is asymptotically stable in large (ASIL).

Since the input is also bounded for all $n \geq 0$, the theorem mentioned in the Appendix of [1] can be applied. It states that for a system described by the state equation $\underline{x}(n + 1) = \underline{A}\ \underline{x}(n) + \underline{B}\ u(n)$, if the homogenous system is ASIL and has a Lyapunov function $\underline{V} = \underline{x}^T\ \underline{Q}\ \underline{x}$ with $\Delta V = -\ \underline{x}^T\ \underline{C}\ \underline{x}$ and $|u(n)| \leq k_1$ for all $n \geq 0$, then the system is stable and the states are certain to enter a region defined by $||\underline{x}|| \leq r_2$, where

$$r_2 = K_1 \sqrt{\frac{\lambda \max(\underline{Q})}{\lambda \min(\underline{Q})}} \cdot \left[\frac{||\underline{A}^T \underline{Q} \underline{B}||}{\lambda \min(\underline{C})} + \sqrt{\frac{||\underline{A}^T \underline{Q} \underline{B}||^2}{\lambda^2 \min(\underline{C})} + \frac{\underline{B}^T \underline{Q} \underline{B}}{\lambda \min(\underline{C})}} \right] \tag{C.11}$$

with

$\lambda \min(\underline{Q})$ = minimum eigenvalue of matrix $\underline{Q}$;

$\lambda \max(\underline{Q})$ = maximum eigenvalue of matrix $\underline{Q}$;

$||\underline{A}^T \underline{Q} \underline{B}||$ = norm of the matrix product $\underline{A}^T \underline{Q} \underline{B}$ defined as max a_{ij} where a_{ij} are elements of $\underline{A}^T \underline{Q} \underline{B}$;

$||\underline{x}||$ = norm of the state vector.

The Lyapunov function $V = \underline{x}^T \underline{Q} \underline{x}$ where $\underline{Q}$ is a real symmetric and positive definite matrix (RSPDM) can be found for any RSPDM $\underline{C}$ from the equation

$$-\underline{C} = \underline{A}^T \underline{Q} \underline{A} - \underline{Q} \tag{C.12}$$

in this case

$$\underline{Q} = \begin{bmatrix} q_{11} & q_{12} \\ q_{12} & q_{22} \end{bmatrix}$$

and since the choice of $\underline{C}$ is arbitrary as long as it is RSPDM choose $\underline{C}$ equal to a 2 x 2 identity matrix. Then from equation (D.12) results

$$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & -b \\ 1 & -a \end{bmatrix} \begin{bmatrix} q_{11} & q_{12} \\ q_{12} & q_{22} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -b & -a \end{bmatrix} - \begin{bmatrix} q_{11} & q_{12} \\ q_{12} & q_{22} \end{bmatrix} \tag{D.13}$$

whose solution is

$$q_{11} = 1 + \frac{2b^2(1+b)}{(1-b)[(1+b)^2 - a^2]}$$

$$q_{12} = \frac{2ab}{(1-b)[(1+b)^2 - a^2]} \tag{D.14}$$

$$q_{22} = \frac{2(1+b)}{(1-b)[(1+b)^2 - a^2]}$$

Defining $\omega = ||\underline{A}^T \underline{Q}\ \underline{B}||$

$$= \begin{bmatrix} 0 & -b \\ 1 & -a \end{bmatrix} \begin{bmatrix} q_{11} & q_{12} \\ q_{12} & q_{22} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} -b\ q_{22} \\ q_{12} - aq_{22} \end{bmatrix}$$

then

$$\omega = \max(|b\ q_{22}|,\ |q_{12} - a\ q_{22}|)$$

and substituting into equation (D.11)

$$K_1 = 1/2 \qquad \text{since } |u(n)| = |e(n)| \leq .5$$

$$\lambda\ \min(\underline{C}) = 1$$

$$\underline{B}^T\ \underline{Q}\ \underline{B} = [0 \quad 1] \begin{bmatrix} q_{11} & q_{12} \\ q_{12} & q_{22} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = q_{22}$$

the following state bound is obtained

$$|x^*(n)| \leq 1/2 \sqrt{\frac{\lambda\ \max(\underline{Q})}{\lambda\ \min(\underline{Q})}} \cdot \left[\omega + \sqrt{\omega^2 + q_{22}}\right] \qquad \text{(D.15)}$$

Comparing equation (D.15) with the one derived by S.R. Parker and Hess [1] for QAM, it can be concluded that the upper bound on amplitude of the limit cycles for quantization after addition is two times smaller than for quantization after multiplication.

LIST OF REFERENCES

1. Parker, S. R. and Hess, S. F., "Limit-Cycle Oscillations in Digital Filters," IEEE Transactions on Circuit Theory, v. CT-18, no. 6, pp. 687-697, November, 1971.

2. Parker, S. R. and Hess, S., "Canonic Realizations of Second-Order Digital Filters Due to Finite Precision Arithmetic," IEEE Transactions on Circuit Theory, v. CT-19, no. 4, pp. 410-413, July, 1972.

3. White, S. A., "Introduction to Implementation of Digital Filters," Electronics Research Division Rockwell International, Publication X73-371/501, March, 1973.

4. Jackson, L. B., Kaiser, J. F. and McDonald, H. S., "An Approach to the Implementation of Digital Filters," IEEE Transactions on Audio Electroacoustics, v. AU-16, pp. 413-421, September, 1968.

5. Jackson, L. B., "On the Interaction of Roundoff Noise and Dynamic Range in Digital Filters," Bell System Technical Journal, v. 49, pp. 159-184, February, 1970.

6. Jackson, L. B., "Roundoff-Noise Analysis for Fixed-Point Digital Filters Realized in Cascade or Parallel Form," IEEE Trans. Audio Electroacoust., vol. AU-18, pp. 107-121, June, 1970.

7. Slaughter, J. B., "Quantization Errors in Digital Control Systems," IEEE Transactions on Automatic Control, v. AC-9, pp. 70-74, January, 1964.

8. Johnson, G. W., "Upper Bound on Dynamic Quantization Error in Digital Control Systems via the Direct Method of Liapunov," IEEE Trans. Automat. Contr., vol. AC-10, pp. 439-448, January, 1965.

9. Lack, G. N. T., "Comments on 'Upper Bound on Dynamic Quantization Error in Digital Control Systems via the Direct Method of Liapunov'," IEEE Trans. Automat. Contr. (Corresp.), vol. AC-11, pp. 331-333, April, 1966.

10. Claasen, T. A. C. M., MecKlenbrauker, W. F. G., and Peek, J. B. H., "Second-Order Digital Filter with only One Magnitude Truncation Quantizer and Having Practically No Limit Cycles," Electronic Letters, v. 9, no. 22, November, 1973.

11. Claasen, T. A. C. M., Mecklenbrauker, W. E. G., and Peek, J. B. H., "Some Remarks on the Classification of Limit Cycles," Phylips Research Report, V. 28, no. 4, August, 1973.

12. Golden, R. M., "Digital Filter Synthesis by Sampled-Data Transformation," IEEE Transactions on Audio and Electroacoustics, v. AU-16, no. 3, pp. 321-329.

13. Naval Electronics Laboratory Center, Report 2570, "Digital Filter Design Utilizing Large Scale Integration Technology," by R.E. Johnston, December, 1973.

14. Blackman, R. B., and Tukey, J. W., The Measurement of Power Spectra from the Point of View of Communications Engineering, Dover Press, 1959.

15. Gold, B. and Rader, C. M., Digital Processing of Signals, McGraw-Hill, Inc., 1969.

16. Kaiser, J. F.,"Digital Filters"in System Analysis by Digital Computers by J. F. Kaiser and F. F. Kuo, Eds., New York, Wiley, pp. 218-285, 1966.

17. Fettweis, A., "On the Connection Between Multiplier Word Length Limitation and Roundoff Noise in Digital Filters," IEEE Transactions on Circuit Theory, V. CT-19, no. 5, pp. 486-491, September, 1972.

18. Aggarwal, J. K., "Input Quantization and Arithmetic Roundoff in Digital Filters - A Review", Information Systems Research Laboratory, Technical Report no. 130, July, 1972.

19. Knowles, J. B. and Olcayto, E. M., "Coefficient Accuracy and Digital Filter Response," IEEE Transactions on Circuit Theory, v. CT-15, no. 1, pp. 31-41, March, 1968.

20. Yakowitz, S. and Parker, S. R., "Computation of Bounds for Digital Filter Quantization Errors," IEEE Transactions on Circuit Theory, v. CT-20, no. 4, pp. 391-396, July, 1973.

21. Mitra , S. K. and Sherwood, R. J., "Estimation of Pole-zero Displacements of a Digital Filter Due to Coefficient Quantization", IEEE Transactions on Circuits and Systems, v. CAS-21, no. 1, January, 1974.

22. Weinstein, C. and Oppenheim, A. V., "A Comparison of Roundoff Noise in Flotaint Point and Fixed Point Digital Filter Realizations," Proceedings of the IEEE (Letters), v. 57, pp. 1181-1183, June, 1969.

23. Oppenheim, A. V. and Weinstein, C. J., "Effects of Finite Register Length in Digital Filtering and the Fast Fourier Transform," Proceedings of the IEEE, v. 60, no. 8, pp. 957-176, August, 1972.

24. Liu, B., "Effect of Finite Word Length on the Accuracy of Digital Filters - A Review," IEEE Transactions on Circuit Theory, v. CT-18, pp. 670-677, November, 1971.

25. Girard, P., "Correlated Noise Effects of Structure in Digital Filters", PH. D. Thesis, Naval Postgraduate School, Monterey, California, 1974.

26. Steiglitz, K., "Computer Aided Design of Recursive Digital Filters", IEEE Transactions on Audio and Electroacoustics, v. AU-18, no. 2, p. 123, June, 1970.

27. Avenghaus, E., "A Proposal to Find Canonical Structures for the Implementation of Digital Filters with Small Coefficient Word Length", Nachrichtentechnische Zeitung, Heft 8, 1972.

28. White, S. A. and Graske, R., "Digital Filter Laboratory Unit", Electronics Research Division Rockwell International, Report T73-218/501, August, 1973.

29. Rabiner, L. R., and others, "Terminology in Digital Signal Processing," IEEE Transactions on Audio and Electroacoustics, v. AU-20, pp. 332-337, December, 1972.

30. Gabel, R. A., "A Parallel Arithmetic Hardware Structure for Recursive Digital Filtering", IEEE Transactions on Acoustics, Speech and Signal Processing, v. ASSP-22 no. 4, August, 1974.

31. G - AE Concepts Subcommittee on Digital Filtering, IEEE Transactions on Audio and Electroacoustics, v. AU-16, no. 3, September, 1968.

32. Parker, S. R. and Yakowitz, S., "A General Method for Calculating Quantization Error Bounds in Fixed Point Multivariable Digital Filters", to be published in IEEE Transactions on Circuit and Systems, April, 1975.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Direcção Do Servico De Instrução Ministerio Da Marinha Lisboa2, Portugal	2
4.	Department Chairman, Code 52 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	2
5.	Professor S. R. Parker, Code 52Px Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
6.	Assoc. Professor V. M. Powers, Code 52Pw Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
7.	LT Carlos J. A. R. Rodolfo Calçada Do Cardeal 24-2-Dt Lisboa2, Portugal	2
8.	Assoc. Professor M. L. Cotton, Code 52Cc Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
9.	Assoc. Professor R. Panholzer, Code 52Pz Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
10.	LCDR L. Souchon, Federal German Navy SMC 2988 Naval Postgraduate School Monterey, California 93940	1

11. Professor J. K. Aggarwal — 1
Department of Electrical and Electronic
Research Center
The University of Texas
Austin, Texas 78712

12. Prof. Bede Liu — 1
Department of Electrical Engineering
Princeton University
Princeton, New Jersey

13. Mr. W. J. Dejka, Code 4300 — 1
Advanced Modular Concepts Division
Naval Electronics Laboratory Center
San Diego, California 92152

14. Dr. G. M. Dillard, Code 3300 — 1
Decision and Control Technology Division
Naval Electronics Laboratory Center
San Diego, California 92152

15. Professor Alfred Fettweis — 1
Rucha University Bochum
Buscheystrasse
D-463 Bockum
Federal German Republic

16. Dr. L. B. Jackson — 1
Rockland Laboratories Division
Rockland Systems Corporation
131 Erie Street, East
Blauvelt, New York 10913

17. Dr. J. F. Kaiser — 1
Bell Telephone Laboratories, Room 7c-201
Murray Hill, New Jersey 07974

18. Professor B. Leon — 1
Department of Electrical Engineering
Purdue University
Lafayette, Indiana 49707

19. Professor S. K. Mitra — 1
Electrical Engineering Department
University of California, Davis
Davis, California 95616

20. Professor A. V. Oppenheim — 1
Room 36-377
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

21. Dr. S. A. White 1
Autonetics Division
North American Rockwell Corporation
P. O. Box 3669
Anaheim, California 92803

22. LCDR P. E. Girard, USN 1
8071 Wildflower Way
San Diego, California 92120

23. Assoc. Professor G. A. Rahe, Code 52Ra 1
Department of Electrical Engineering
Naval Postgraduate School
Monterey, California 93940